

Transcribed from the Disk Magazine

COMPUTE!'s GAZETTE

February 1995
Disk Issue No. 14

FOR COMMODORE PERSONAL COMPUTER USERS



Transcription prepared for the CBM 8-Bit Magazine Index by Dr. Myles L. Skinner.
Please visit https://www.tierceron.com/commodore/magazine_index.php for
more information about the index.

GAZETTE DISK MASTHEAD

Tom Netsel, editor

Address editorial matters to the Greensboro office or via E-mail: Gaz on QuantumLink,
TomNetsel@aol.com on the Internet

Sylvia Graham, copy editor

Editorial offices:

COMPUTE's Gazette

324 W. Wendover Ave., Ste. 200

Greensboro, NC 27408

(910) 275-9809

(910) 275-9837 fax

Subscription information:

Gazette Disk

P.O. Box 3244

Harlan, IA 51593

(800) 727-6937

Call this number for any questions about your subscription, change of address, missing or damaged disks.

Advertising information:

Lucille Dennis

(707) 451-8209

Administration:

Kathy Keeton, president and COO

Bill Tynan, executive vice president operations

COMPUTE Publications Intl. Ltd

1965 Broadway

New York, NY 10023

(212) 496-6100

All material on this disk, with the exception of public domain programs featured in "PD Picks," is copyrighted 1994 by COMPUTE Publications Intl. Ltd.

Cover image created by Chris Ryan using *The Print Shop* from Brøderbund Software and originally printed on a Panasonic KX-P1180 9-pin dot-matrix printer.

Transcription prepared for the CBM 8-Bit Magazine Index and edited by Dr. Myles L. Skinner (MS). Please visit https://www.tierceron.com/commodore/magazine_index.php for more information about the index.

Table of Contents

64/128 View: Time to Say Goodbye <i>Tom Netsel</i>	3
Advertising	49
Columns	
D'Iversions: Undigitize Your Presentations for a High-Tech Look! <i>Fred D'Ignazio</i>	5
Machine Language: Lunar Landing <i>Jim Butterfield</i>	9
Beginner BASIC: Column Reprints <i>Larry Cotton</i>	12
Programmer's Page: Money, Money, Money <i>David Pankhurst</i>	14
GEOS: New Products <i>Steve Vander Ark</i>	18
PD Picks: Cosmos Designs and Warriors <i>Steve Vander Ark</i>	20
Features	
Hints from a Heavy User <i>Don Radler</i>	22
Steel Trap <i>Larry Cotton</i>	28
Review: geoCanvas 3.0 <i>Steve Vander Ark</i>	32
Programs	
Satellite Tracker (for the 64 and 128) <i>Donald G. Klich</i>	34
ScreenSaver (for the 64 and 128) <i>Martin Fensome</i>	35
Display Maker <i>Tom Zdanowicz</i>	37
Mapmaker <i>David Pankhurst</i>	43
Mailing List <i>B. A. Carwithen</i>	46
Hurricane <i>David Garner</i>	48
Ecstatic Code <i>Cosmos Designs</i>	48
Warriors <i>Paul Spomer, Marc Turner, and David Turner</i>	48

64/128 VIEW: Time to Say Goodbye

Tom Netsel

An era comes to an end. This is the final issue of Gazette.

In July of 1983, a new magazine devoted to the Commodore VIC-20 and the Commodore 64 hit the newsstands. Founded by Robert Lock and edited by Thomas Halfhill, that magazine was called *COMPUTE!'s Gazette*. After almost 12 years, numerous editors, and several changes in format, *Gazette* ceases publication with this issue. This is the last *Gazette*.

When *Gazette* started, Commodore was selling more than 100,000 VIC-20s and 64s each month. Dozens of software suppliers were rushing products to this rapidly growing market. The initial press run for *Gazette* was 175,000 copies, up from an originally planned 75,000. And all this was at a time before the 64 itself really took off to become the most popular home computer ever built, selling more than 14 million worldwide.

If you look back at that first issue, you'll notice that Fred D'Ignazio had a column that was called "Computing for Kids." Jim Butterfield's name appeared on the masthead as an associate editor. You'll find their final columns in this issue.

Larry Cotton is another long time contributor. His "Beginner BASIC" column has been a regular feature for many years.

A few years ago, I was looking for someone to write a GEOS column. It had been an on and off feature in *Gazette*, but readership surveys told me that there was considerable interest this operating system from Berkeley Softworks. I went searching for someone to write a monthly column. I'll have to thank Robert Nellist for recommending Steve Vander Ark for the job. Robert and Steve had been corresponding for some time about GEOS, and he suggested I get in touch with Steve. I did and Steve agreed to supply *Gazette* with a column each month.

Sometime later, I wanted a column to pay tribute to the great work being done by programmers who offered their work as public domain. Steve volunteered to do it and added "PD Picks" to his busy schedule. It was about that time that Robert wrote me a letter complaining in a good natured fashion that while *Gazette* had gained a columnist in Steve, Robert had lost a pen pal.

The junior member of the *Gazette* columnists is David Pankhurst. The author of fantastic spreadsheets, *Calc* and *Calc II*, David took over the reins of "Programmer's Page" a little more than a year ago. Each month he has supplied a number of good programming tips from his own library and from those submitted by our readers.

I'd like to thank all these gentlemen for their great contributions to *Gazette*. Without their help, this magazine couldn't have lasted a fraction of the time it did.

Last, but by no means least, I want to thank all of you readers who have submitted programs to *Gazette* over the years. My biggest regret is that I still have so many good programs here that I'll never be able to publish. My budget allowed me to buy only so many each month and I still have almost 30 submissions still in my Pending basket. *Gazette* had a number of staff programmers in its early days, but it has been our readers who have supplied the bulk of the

programs that we've published over the years. To all of you who have submitted a program, I offer you my thanks. Without your submissions, this magazine would have folded years ago.

Gazette's closing follows shortly on the heels of *COMPUTE's* demise, but it is for totally different reasons. *COMPUTE* was bought and closed because a competitor didn't want the competition and was hoping to convert those subscribers to its publications. *Gazette* is closing simply because there aren't enough Commodore enthusiasts subscribing to make *Gazette* profitable.

So what's going to happen to your subscriptions? You'll have one of three options, but as I write this the week before Christmas, I don't know all the details as yet. We are talking with other publishing companies that may have an interest in taking on our outstanding subscriber liability. If one of these companies takes over, then you will be offered a Commodore-related publication equal in value to the number of *Gazette* issues yet unmailed.

The other option is that you may elect to take the remainder of your subscription out in *Gazette* disk products. The specialty disks that we offer, such as *SpeedScript* and the GEOS Collection, are examples.

The final option is to simply request a refund of the remainder of your subscription. Each unmailed issue is worth approximately \$2.50.

As I said, negotiations have just started and I don't know what the outcome will be. You will be contacted shortly with an explanation of your options. I'm sure some of you will be getting letters asking you to renew your *Gazette* subscriptions. These mailings are scheduled months in advance and some undoubtedly will be mailed out after I write this. Please ignore them.

Normally, publications close and readers learn about it after the fact. With *Gazette* being on disk and having a short production time, I was able to convince the powers that be to give me this final issue in order to say goodbye. As I said, I don't yet have all the answers but I was at least able to offer some word of explanation as to why you won't be getting any more *Gazettes* after this one.

COMPUTE Publications will still be around, but involved with online publishing. *COMPUTE Online* has been on America Online for 18 months and it will continue, but with a new name. Starting in February, it will be called *I-Wire*. It will also be on the Internet with a world wide web home page. I have been the online editor during this time and I will continue with my duties there and as editor of *Saturday Review Online*.

Once again, thank you for your support over the years. Keep those 64s and 128s humming. It's been a pleasure.

D'Iversions

Fred D'Ignazio

Undigitize Your Presentations for a High-Tech Look!

I fly around the country giving speeches and multimedia demonstrations in big auditoriums, hotel rooms, and convention centers. When I arrive on location I spend anywhere from four to eight hours setting up my equipment to make a presentation. Sometimes I speak to only 30 or 40 people, but on occasion I've spoken to as many as 6000 to 8000 people.

I also do television, having appeared three times this past fall. And when I do, I set up the exact same equipment—usually on an eight foot table. On top of the table are as many as 50 tiny boxes and devices. Under the table looks like the eerie twilight of a rain forest, with every species of cable hanging from the table like big boa constrictors and long Tarzan-tempting jungle vines.

Road Warrior

I've managed to fit the 700 items in my on-the-road mobile multimedia presentation kit into only four big Samsonite cases which I've all fondly named—Tootsie Roll, Wendy Burger, Octopus, the Brick—based on their shapes.

Two of the bags have wheels so when I'm on the road, I stack the other cases (and various mini-bags which I smuggle aboard airlines by several patented devious maneuvers) on top the wheeled roll-aboards. I drag the giant Wendy Burger roll-aboard behind me and push the Brick ahead of me. Then I set off through an airport or hotel and don't come to a stop until I hit a wall, a potted plant, or a steep looking escalator.

Fellow travelers in the airport see me coming from far away and usually give me wide berth. To help them prepare for my arrival, I go through the airport chanting "Road Warrior! Road Warrior!" This phrase keeps me focused and usually guarantees me an open pathway through the congested, pedestrian gridlock of most airport corridors.

And for Only \$24.95

What makes this amazing array of multimedia equipment come together? A small switcher box—namely the Archer audio/video selector switch, catalog #15-1956—which costs only \$24.95 at your local Radio Shack store. When I explain this miraculous box to my audiences, I liken it to an open funnel, starting fat at the top, then narrowing down to a little tip at the bottom. Feeding into the switcher box are four sets of audio (two for stereo) and video cables. These cables typically come from the following sources.

1. My laserdisc player.
2. My VHS VCR.
3. My camcorder.
4. My computer.

With the mere click of a button, I can select any of these four audio-visual inputs and the selected video source goes through a single output on the back of the box and up onto an LCD panel on my overhead projector and from there onto a big screen, sometimes nine feet

wide! (Also, I've begun daisy-chaining TV monitors on high carts to help members of the audience who sit on the far right or left and who can't easily see the big screen.)

When I press the switcher button, the selected audio goes from the switcher to a small, portable Roland speaker (which I bought at a local music store where they sell supplies to electronic musicians.) This speaker has a headphones jack which can easily be mated to a P.A. system to get big sound.

Multimedia DJ

When I'm making a presentation, I keep the room in complete darkness. I move around my presentation table with the help of a small pocket flashlight, which I keep pointed down at the floor.

Before making a presentation, I cue up all my media—laserdiscs, VHS tapes, 8mm tapes (shown on the tiny camcorder, mounted on a desktop mini-tripod), and computer slides. I lay all the media out on the table before me, labelled side up, so I can easily grab an item when I need it. This allows me the freedom of jumping around in my media, at random, based on the audience's questions and interests, and the flow of my presentation as it develops.

It makes me feel like a multimedia disc-jockey putting on the platters that my audience loves me to play. (Of course they don't know what I'm going to play next, but I try to feel the mood in the room, based on their body language and laughter, then I put on what seems right as the next selection. I think this is what a good radio DJ does, too, when he or she is running through a set of music, like jazz, blues, or country music.)

The Secret of Undigitizing

At the end of a multimedia presentation, I love to hear reactions from the audience.

"Fantastic job, Fred! I just loved the way you got all that full-screen, full-motion video! You must have one of those new \$10,000 video circuit cards and a 150-MHz Octahedral (or some such) processor chip, a ReelMagic video card, a 5MB video cache, and 64 MEGs of RAM to push the video through the computer as fast as you do! Great show!"

At that point I can't tell you the satisfaction I get when I point to my \$25 Radio Shack switcher and tell them I did it all with that little box. How do I create the illusion of full-frame, full-motion video with a tiny passive (non-powered) switch box?

The trick is in the switching. As one high-tech savant once said: The magic is not the meat, it's the motion!

When the room is dark, it sets everyone's imagination free. They've all seen my table with all its tiny boxes and wires. They look up at the swift procession of images on the big screen and they hear the big, booming sound coming from the P.A. speakers. And they just imagine that I'm doing all this cool, expensive multimedia magic at my Alchemist's workbench.

Actually all I'm doing is clicking four little buttons. The first is to the computer to show a title, a graphic, and maybe some key subject. Then, click, and I'm showing a clip from the laserdisc, then, click, I'm showing a segment from a VHS tape, then, click, and I'm playing a piece from my tiny camcorder.

The trick is that people imagine that I'm showing everything through my computer. That is, they think I've digitized all my media, and they marvel at the high resolution full-motion video and the CD-quality sound.

"The colors!" They exclaim, when they see a National Geographic tape of an Australian rain forest. "They're so vivid, so vibrant, so real!"

"The power, the motion!" they cry, as they watch the Apollo astronauts blast off on a big Saturn rocket, as I play my Voyager videodisc "For All Mankind."

After the lights come on at the end of my presentation, people swarm around my workstation and ask me a flood of questions:

"How did you make the video play without any jumping, burps, or jerky hesitation?"

"How did you keep your audio track so clean, so clear, and keep it synthesized exactly with the video?"

"How did you get all those video and audio files off your hard drive so quickly? They seemed to leap onto the screen almost as soon as you clicked on your buttons?"

Again, I smile with enormous satisfaction, and I point proudly down at my little switch box. "The secret is in this box," I say quietly. "I do it all by UNdigitizing."

The Moral of the Story

The moral of this story is don't digitize unless you have to.

When you try to digitize all your media, you are sure to put a big load on your computer. Digitizing forces your computer to work much harder. It means you need tons of extra CPU power, RAM, gobs of hard disk space, and all kinds of special circuit cards, buses, caches, motherboard enhancements, and so on.

Most non-techie people are used to TV. TV is produced by networks who have million-dollar, special-purpose boxes that make graphics a high art form akin to magic. It is highly probable that you will not be able to duplicate this real-time, super-dazzling media on your desktop computer, no matter what kind of computer you have.

The average person is not concerned with technical details (such as throughput). All they want is what TV has been delivering for years: instant full-screen, full-motion video, with surround-sound speakers, just like you get in the movie theaters and in some living-room entertainment centers.

So don't bother trying to soup your computer up to try to please them. If you do, you might end up going broke.

Instead: UNdigitize! Try to pass your audio and video sources directly from the source to the presentation device (TV, LCD panels, speakers, screen, and so on) using a simple little switcher.

After all, does it really make sense to DIGITIZE all that stuff into your computer, just to turn around again and UNdigitize it in order to get it out of the computer to present to your audience?

This might be heresy in some power computer quarters, but I'm going to end this article with the following advice:

Sometimes it's ok to skip your computer when you are making a presentation. Your output may be more impressive and you may end up saving LOTS of money!

**

Postscript

(This postscript appears twice on the disk: once appended to the article itself, and again in a separate file. There are some small, subtle differences between the two versions, so I have reconstructed a single, coherent version here. Please consult the .d64 to see the original versions. —MS)

A Few Final words: I have been writing for *Compute!* or *Compute's Gazette* on and off (mostly on!) for almost 15 years—ever since I first spoke to the original *Compute!* founders in Greensboro, back in the late 1970s.

Over the years I have watched *Compute* and its sister magazine the *Gazette* grow from thin little pamphlets reaching a few hundred people into major magazines read by (in their heyday) hundreds of thousands of home computer enthusiasts.

Compute and the *Gazette* have always given me complete freedom to write exactly what I felt like. As a result, over the years, the editors (like wonderful Tom Netsel!) have had a constant supply of material shuttled to them directly from a pretty wacky subconscious. Over the years my *Compute* and *Gazette* readers have met a parade of zany UFOs (Unimaginable Fred Objects) including: Frank the Fireplace, the Computer Friend, Sticky Tab Computers, Santa, Subways, and Penguins, Lobster Robots, and the like.

The connection between my imagined worlds and computers might have sometimes seemed far-fetched, but you readers never seemed to mind. And you allowed me to stay there in my little imaginary control tower, scanning the world with my binoculars looking for new amazing objects to describe and relate to computers.

But now our time together is over, so, sadly, I must tell you good-bye. But before I do, I want to say that I am proud to have been serving you all these years, acting as one tiny kowabunga spokesperson surfing the giant tidal wave of personal computing along with the rest of you.

(Wow! What a ride!)

And since you buy this disk and read this magazine, I think we probably believe the same things. We believe that computers are so neat they are almost magical. And they are magical like puzzles and riddles and mysteries so we want to investigate them, peer inside them, take them apart, grow small in our imaginations and take a fantastic voyage through the marvelous world that lies inside each and every computer.

And another shared feeling that we share deeply: Long live older computers! Especially old C=64s. But also old C=128s and old Vic-20s. And even old Atari 400s and 800s. And old TI-99/4As. And old PETs and old Z-80s and old ThinkerToys and old Apple Is and IIs and IIIs and ... They are all wonderful machines and I hope anyone who owns one of these machines and keeps that little baby humming.

With a little imagination and a sense of wonder and curiosity, older computers can serve us for years for our hobbies, our work, and for some good clean hacking.

Last, I think we share the same awareness of the funny, kooky side of the computing scene. The “interface” between people and computers has been the neighborhood of my columns for almost fifteen years. The strange, wonderful, and weird things we humans do (or might do!) when we use these machines have fascinated me, and I hope they have fascinated you as I write about them each month.

So, good-bye, my friends. Don’t throw your beautiful old computers away. I’m sure you can dream up at least one more creative way to keep using them. And if you can’t, donate them to a toddler. Toddlers simply LOVE old computers!

Your computer friend,
Fred

P.S. Whisper: Sshh! This is a secret! If you REALLY haven’t had enough of me, then I invite you to write me, and we will keep this column alive interactively through correspondence. My e-mail address on America Online is Explorer00. On internet you can reach me at Explorer00@aol.com or at dignazio@msu.edu.

Please write me. I look forward to hearing from you! Bye for now.

“Phhhhttt!”

(Fred dematerializes, looking a little sad, a little awestruck, and a little bewildered. But just before he fades away, he suddenly grins, waves wildly, and flashes the victory sign!)

Machine Language

Jim Butterfield

Lunar Landing

The equations of motion for a lunar lander are not complex. We’ll put together a neat instrument-only program that will have you touching down in no time.

Here’s the math. Thrust is whatever power you set in your rockets; you’ll do that with keys 1 to 9.

Acceleration is thrust minus the force of gravity; we’ll set gravity to a value of 5, making acceleration equal to thrust minus 5; note that this value will be negative if gravity is the main force.

Velocity increases or decreases according to acceleration, but your velocity had better be low when you land. Again, this value could be positive or negative, depending on whether you're going up or down.

Finally, height increases or decreases according to velocity.

There's one more calculation, not directly related to motion. Fuel decreases according to thrust. And when you're out of fuel, the value of thrust will drop to zero, no matter what key you press.

You'll start at a height of 4500 feet. There's plenty of fuel to get you down safely, if you have a steady hand on the controls. Machine language is your control linkage, but you're the pilot.

A Little More Math

The above description uses the term "increases or decreases according to...". Let's be more specific. Every 1/100 second, we will add the acceleration divided by 100 to the velocity. When a second has elapsed, the velocity will have changed by the amount of the acceleration; 100 additions will have taken place.

The same type of calculations will take place to have velocity influence height, and thrust influence the fuel supply. The resulting values, sent to the screen, change smoothly as you navigate your flight.

How do we divide a value by 100? Easy, especially in decimal mode. We just add a couple of digits to the value, giving a fractional part. Thus, an altitude of 4500 feet is held as 450000: three bytes of data, 45, 00, and 00. To add in a velocity of, say 25, we add to the fractional part, giving 450025 or a height of 4500.25 feet. Of course, we display only the 4500.

How do we measure time intervals of 1/100 second? Well, we cheat a little on this one. The video raster takes 1/60 second to draw the screen. By performing two calculations during each raster sweep, we actually do 120 of them per second; close enough for this flight. In Europe and other PAL-display locations, the timing is 1/50 of a second per raster, which would be exact.

There's another reason for choosing decimal mode calculations. Our main job is delivering the numbers to the display, and that's easier to do in decimal mode. Binary numbers would call for us to perform conversion of each value.

Coding Details

Initialization is a simple matter of setting values for altitude, fuel, and so on. A simple loop does the job. Then we move along to our main task of calculating and displaying flight information.

Our first task in the main loop is to wait for the raster retrace; this is the familiar job of checking address \$D012. Then we start to consider thrust information from the keyboard.

Before we test the keyboard, a quick look at the fuel gauge. If we're out of fuel, thrust will be zero and keyboard input doesn't matter. There's a gimmick used for the empty fuel condition, that saves us testing time.

If we were to test the fuel value for zero, we'd need to check all three bytes. It's more efficient to allow the fuel value to go negative: in decimal mode, this would be a value such as 9999.99. Now we can simply test the high byte (its high bit!) to see that the tank is dry. As soon as we see this, we reset the fuel value to hexadecimal F0 00 00, so that the value will display as zero, but the negative test for out-of-fuel will still be valid.

If we do have fuel, we check to see if a key has been pressed. If a key in the range 1 to 9 is found, we log it as the new thrust value, in both binary and ASCII. Thrust 0, by the way, would turn the motor off, and it won't re-ignite.

Flight Calculations

We've outlined the nature of the flight calculations. There's little detail to be added, except that negative numbers are taken care of automatically. Keep in mind that in decimal mode, negative 1 is represented as 9999 (not \$FFFF as in binary).

That works fine for the calculation part, but wouldn't look too good on the pilot's display panel. We'll show how to handle that in a moment.

It's worth noting that the highest positive value that can be held in decimal mode is 7999. The lowest negative value is 8000, equivalent to -2000. So (at least using the high bit to indicate sign) you have a greater range of positive values than negative.

Instrument Display

The instrument values are poked directly to the screen. That saves us the work of setting cursor positions, and so on as would be needed if we had chosen to use CHROUT (\$FFD2). But it means that the 128 cannot use its 80-character screen for this program.

A subroutine takes the decimal mode (BCD) values and converts them to four ASCII characters. We do not need to print all four: velocity and fuel reading are three digits only.

Velocity may be a negative value. The display routine first carefully sets the "+" or "-" character. Then, if the reading is negative, it is converted to absolute value: 9999 becomes -1. We do this by subtracting the value from 0, in decimal mode, of course.

We detect a landing when we see the altitude go to a negative value. At that time, the BASIC program takes over. It peeks the velocity from the screen display, and decides how good a pilot you are.

Note that you are not allowed to go above 7999 feet in altitude. Do you know why?

Flight Plans

If you have never flown this kind of craft before, here's a hint. Get down fast, or you might run out of fuel. But not too fast, or you'll dig a hole when you land. If you're an experienced pilot, try these challenges. First: can you arrange to land safely with 0 fuel? Second: can you level off exactly at 5000 feet, and then get down safely?

On the flip side of this disk you will find Lunar programs for both the 64 and 128, plus the source code that you can load and list.

Beginner BASIC

Larry Cotton

Column Reprints

One advantage to *Gazette's* new disk format is that, over a period of years, you're more likely to keep up with all the back issues, since they take up so little space. However, from July, 1983 through December, 1993, *Gazette* was a magazine. Anyone who's kept all the back issues has almost a three-foot stack. (Trust me; I know).

So this month, I'm going to make you an offer you can't refuse. No, I'm not trying to unload any swamp land in Florida. However, I do have, on several disks, all the "Beginner BASIC" columns and their associated programs since I began writing them way back in October, 1986. (Back then it was called "BASIC for Beginners.")

Need to bone up on DEF FN? Need to use a joystick port for output? Or how about the ultimate crisis: you've got a kid who's having trouble with trig.

"Beginner BASIC" to the rescue! Here's the deal: I'll be happy to make you a copy of up to three columns—and their associated programs, if any—for free. All you have to do is tell me which column(s) you'd like and send me a 5 1/4" Commodore-formatted disk along with a self-addressed, stamped disk mailer. The articles are written in *Speedscript*, and the programs are ready to run.

Or, if you're feeling flush, send me ten bucks and I'll let you have ALL the columns. That's well over 100 articles and programs packed on four double-sided 5 1/4-inch disks!

You'll need a handy month-by-month index, so here it is.

Oct 86 Introduction and PRINT
Nov 86 PRINT
Dec 86 PRINT and INPUT
Jan 87 FOR-NEXT
Feb 87 IF-THEN
Mar 87 IF-THEN, cont.
Apr 87 GOTO and GOSUB
May 87 ON
Jun 87 GET
Jul 87 State Capitols
Aug 87 READ and DATA
Sep 87 RESTORE and TAB
Oct 87 TAB, SPC and Strings
Nov 87 MID\$
Dec 87 LEN, VAL STR\$
Jan 88 Unstructured Programming
Feb 88 RND
Mar 88 RND, cont.
Apr 88 Math Drill Program
May 88 Math Drill Program, cont.
Jun 88 My Dear Aunt Sally

Jul 88 Basic Geometry
Aug 88 Rounding
Sep 88 Arrays
Oct 88 Arrays--Smalltown 500
Nov 88 Variables Revisited
Dec 88 Wish List, Disk BASIC
Jan 89 More Disk BASIC
Feb 89 FN and DEF FN
Mar 89 CHR\$, E, large and small
numbers, calculated array subscripts &
metric conversion
Apr 89 Metric Converter
May 89 E
Jun 89 Bugbusters
Jul 89 Musical POKEs
Aug 89 More Musical POKEs
Sep 89 Nintendo vs C64
Oct 89 Good Vibes
Nov 89 Keyboard
Dec 89 Keyboard—Playing It
Jan 90 Lesser Known Commands

Feb 90 Lesser Known Commands, cont.
 Mar 90 AND, OR, NOT
 Apr 90 Joysticks
 May 90 Timers
 Jun 90 Timers, cont.
 Jul 90 No issue published
 Aug 90 No issue published
 Sep 90 No issue published
 Oct 90 Musical Sprites
 Nov 90 Musical Sprites, cont.
 Dec 90 NOT and Multiple Sprites
 Jan 91 Multiple Sprites, cont.
 Feb 91 This Program Won't Run
 Mar 91 Request for Reader Input
 Apr 91 Digital Dialer
 May 91 Digital Dialer, cont.
 Jun 91 BASIC Math
 Jul 91 Trigonometry
 Aug 91 Solving Right Triangles (includes complete program)
 Sep 91 Crashproof Data Entry
 Oct 91 Crashproof Data Entry, cont.
 Nov 91 RND (General)
 Dec 91 RND (Twinkling Tree)
 Jan 92 RND (Music Patterns)
 Feb/Mar 92 RND (Music Patterns End)
 Apr 92 Keyboard Buffer (Colorcalc)
 May 92 Amortization
 Jun 92 Transfer Char. Data to Sprites
 Jul 92 Joystick Output (LED)
 Aug 92 Joystick Output (Relay)
 Sep 92 Keyboard (Function Keys)
 Oct 92 Keyboard (Other Keys)
 Nov 92 Challenge: BASIC vs ML RND

Dec 92 GET
 Jan 93 INPUT
 Feb 93 WAIT/LET
 Mar 93 FOR-NEXT
 Apr 93 STEP/German Magazine
 May 93 Summary of Nov 92 Challenge
 Jun 93 User Port Clock
 Jul 93 User Port Clock, cont.
 Aug 93 FOR-NEXT nested/Derek's problem
 Sep 93 READ/DATA/INPUT
 Oct 93 PRINT#, INPUT#, start TODOLIST
 Nov 93 PRINT#, INPUT#, finish TODOLIST
 Dec 93 Challenge: Sets, Rounding, Westminster Chimes
 Jan 94 Custom Characters
 Feb 94 Custom Characters, cont.
 Mar 94 Lessons
 Apr 94 GOTO vs GOSUB
 May 94 CHR\$
 Jun 94 Setz! by Deweese and Moore
 Jul 94 Rounding by Merritt
 Aug 94 Review of Variables & Constants
 Sep 94 Units Converter (Variables, Constants, Arrays)
 Oct 94 Units Converter finished
 Nov 94 Custom BASIC prgs—Financial Calculations
 Dec 94 Custom BASIC prgs—LIST MAKER
 Jan 95 Evolution of a program—SURVEY TAKER

Oh, yes, the address:

Larry Cotton
 3513 Canterbury Rd.
 New Bern, NC 28562

Please allow a few weeks for delivery, and don't forget to put your return address on the stamped mailer!

Postscript

The ultimate PostScript: *Gazette* editor Tom Netsel called today to inform me that this is the last *Gazette Disk*. Although this comes as very sad news to me (and I'm sure also to you), it's truly awesome that the Commodore 64 (and *Gazette*) survived as long as they have in this world of 64-bit computers, gigabyte hard drives, megs and megs of RAM and Super VGA.

From Volume 1, No. 1 in July, 1983, *Gazette* has always provided the highest quality articles, the best type-in programs, and the most authoritative reporting on the Commodore 64 computer. That's why *Gazette* survived almost 12 years while its competitors fizzled and failed.

When I started this column in October, 1986, *Compute!'s Gazette's* 143 pages were packed with many exciting articles, along with an actual Commodore ad that read, "If Personal Computers Are for Everybody, How Come They're Priced for Nobody?" At the bottom of the page were the words "under \$600." (This was for the computer itself, without any means of program or data storage, or a monitor.)

Approximately 100 columns later, I can honestly say I've enjoyed every month's challenge of coming up with a topic that would be educational, interesting and fun—all the while sticking to the Beginner BASIC theme. I'm thankful to the editors for putting up with the occasional typo or missed deadline, and for letting me stick it out to the end.

But hey, there's always a tiny bit of silver in every cloud, so here's a nugget: I'll be right here, pounding away on the 64's keys, mostly for the pure joy of it. If you can't get a BASIC program to run, or you just want to yack about 64's and 8-bit computing in general, drop me a line. If you really do need some help, include your listing, a formatted disk and sufficient return postage. My address is printed above. Heck, I'll even toss in March and April's unused columns on the return disk.

One more thought: I want to thank my wife, Sylvia, for putting up with all my mumblings, grumblings, bumblings and humblings for the past eight years. Although she's never so much as laid a pinky on a computer keyboard, without her support, I would have probably hung up the ol' joystick long before today's phone call.

Take care, stay in touch, and keep those BASIC programs running!

Programmer's Page

David Pankhurst

Money, Money, Money

Although it's a number, it has a style all its own. Everyone wants it, needs it, and depends on it. And it can be quite the bane to programmers. What am I talking about? Money!

With its unique needs, working with dollars and cents can be downright tricky for the programmer. Fortunately, there are plenty of tips and programmers ready to provide them, so let's look at ways of working with money.

Using Sense About Dollars

Dollars have special needs on the computer. You're modelling something from the real world, and the real world takes a dim view of being messed with. Although Commodore BASIC may tell you that 17 percent of 50 cents is 8.5 cents, try telling that to a customer who wants to pay 8 cents, or the taxman, who wants 9 cents (actually, more like 54 cents, but that's another

story, and column). This problem of what to do with the fractions of a cent must be taken into account with every calculation.

If this kind of money rounding error unnerves you, you're not alone. On occasion, the uncertainty of what to do with parts of a penny can lead to real problems. One historic case of computer crime involved a programmer who, when faced with these fractions in an accounting program, programmed the computer to place them into his account. These fractions of a cent accumulated, netting a tidy sum (and a stiff judgment when discovered).

To keep money in perspective when programming, here's a few tips:

(1) There are going to be fractions, so plan in advance how to deal with them. Methods include accumulating them, ignoring them, and rounding all values; each method has distinct advantages and disadvantages.

(2) When calculating results, follow accepted accounting practises. If an accountant would take 15 percent, round to the nearest cent, and then subtract, you do so. If he totals before discounting, do the same.

(3) Remember the Commodore internal format isn't perfect. For instance, 0.01 dollar (one cent) can't be stored exactly, and errors may crop up on large amounts. One solution is to work with pennies, not dollars, and divide by 100 when you need to print out dollar values.

(4) Double check everything with real data. Before you set a program to work on your taxes, make sure the results match with what you got with pencil and paper. This is money, so be even more careful when checking out your program.

What about fractions?

Rounding and truncating are two ways that fractions of a cent are dealt with. The difference between them is that rounding adjusts the results up or down to the nearest cent; truncating lops off the fractional part, meaning the result will never be larger than the original amount (something that can happen with rounding).

Another less common function is one that raises to the next cent if there's any fraction of a cent; using this, \$4.551 would become \$4.56, but \$4.550 would become (or remain) \$4.50.

Thomas Turner, of Waterloo, Indiana, wrote a general purpose rounding function that works well with any number. For money, we're only concerned with rounding cents, so a simplified form is shown here, as well as three other functions to handle money. They're all packaged in a program called *Money Functions* on the flip side of this disk.

```
100 :  
110 REM MONEY FUNCTIONS ON DOLLARS  
120 :  
130 REM NOTE THEY MAY BE INACCURATE ON LARGE NUMBERS DUE TO ROUND-OFF  
140 REM RD(X) ROUNDS UP OR DOWN TO A FULL CENT  
150 REM RS(X) RAISES TO NEXT FULL CENT IF THERE'S A FRACTION OF A CENT  
160 REM TR(X) TRUNCATES, OR LOPS OFF ANY FRACTION OF A CENT  
170 REM FR(X) RETURNS ONLY FRACTION OF A CENT  
180 :  
190 DEF FN RD(X)=INT(X*100+.5)/100
```



```

200 :
210 DEF FN RS(X)=INT(X*100+.9999)/100
220 :
230 DEF FN TR(X)=INT(X*100)/100
240 :
250 DEF FN FR(X)=X-INT(X*100)/100
260 :

```

Besides the three functions mentioned, a fourth function, `FR()`, is included, which strips off the full dollar and cent amount, leaving only the fractional part.

The program consists of two sets of functions. The above ones (from lines 100-260 in the program) deal with dollars, where a cent is 0.01, and a dollar is stored as 1. If you decide to work in cents instead of dollars (where 100 is a dollar, and 1 a penny), the functions following line 260 in the program should be used instead, since they work on this format of money.

Putting Out the Bucks

One weak area of the Commodore is printing numbers. Anyone who has tried to format a column of numbers knows the problems involved. In the real world, numbers are neatly formatted. Look at any check, and you'll often see a dollar sign tucked up against the amount, and a string of asterisks to the left. Other financial features include `()` for negative amounts, DB and CR for credit and debit respectively, and commas separating digits of long numbers.

This number formatting is done in BASIC with the command `PRINT USING`. The 128 has a simple version of that command, which prints numbers based on a formatting string. The 64 isn't so blessed however, so to rectify this problem look for *Number Filter* on the flip side of this disk.

On running the program, you select where you want this relocatable routine to be placed. From there, the `SYS` command to execute it is:

```
SYS [start address],X$,F$,X
```

Where `X$` is the string variable the result is placed into, `F$` is the formatting string or variable, and `X` is the number to format. An example would look like this.

```
SYS49152,N$,"AAAVAA",X
```

This would format `X`, placing the result into `N$`. The routine works by first creating an image of the number in memory, just as if it had been printed. This number is then copied digit by digit into the output variable, using the formatting string as a guide. A character in the format string determines what is placed in that same position in the output string, such as one of the number's digits, or a dollar sign or comma.

The format character codes that perform actions are listed below. Only these characters have any effect; all other characters in the formatting string are copied to the output string unchanged.

A - A digit from the number is placed here. If there are no more leading digits in the number to be copied, a space is placed here.

D - Like A, it prints a digit. However, if there are no more leading digits to copy, zeroes are placed in the string instead of spaces. This command is especially useful for formatting part or serial numbers, where leading zeroes need to be displayed.

Z - If the whole number is zero, then this character is replaced with a space; otherwise, a digit from the number is placed here.

S - The sign of the number replaces this character, either plus (+) or minus (-).

N - Like S, the character is replaced with a sign; however instead of the + sign, a space is used for positive numbers.

V - This character marks the decimal place in the filter, and a period (.) is placed in the string output.

CR - For Credit. These two characters, when together, are copied into the string output variable when the number is positive; otherwise, two spaces replace them.

DB - For Debit. Like CR, these letters or spaces are output depending on the number's sign. In this case, a negative number means DB is copied.

C - A comma is placed here in the string variable if there are more leading digits in the number to be displayed.

M - A floating dollar sign is printed in front of the number.

F - Same as M, but the spaces in front of the dollar sign are filled with asterisks.

P - Floating parenthesis are printed if the number is negative.

Although there isn't enough room to discuss the formatting characters fully, a few examples should help clear things up.

FILTER	NUMBER	RESULT
"AAACAADVDDD"	12345.678	" 12,345.678"
"FFFCFFDVDD"	765.432	"***\$765.43"
"PPPCPPDVDDP"	-2345.891	" (2,345.89)"
"MMCMMDVDDS"	245.12	" \$245.12+"
"ZZZZZZZZZZ"	0	" "
"#DDDDDD CR"	1345.22	"#001345 CR"

Note that all other unrecognized characters are printed as they are, and the formatting process chops off unused digits and signs (as in the examples above), so it's up to you to notice and display them. These format strings should give you ideas of your own; alternately, you can save effort by just reusing the above examples.

What's Your Money Worth?

In conclusion, *Money Queries* is a program that answers several questions about money, such as:

How much money do I need to invest now to get a certain amount later?

I have the money, now how much can I make by investing it?

How long will it take to pay off a loan if my payments are a certain amount?

How big do loan payments have to be to pay them off by a certain time?

The program is menu driven and easy to use, with explanations of all the calculations and results. And since you've made it to the end of this column, you don't need to be told not to expect the results to be accurate to the fraction of a cent!

Money Queries, on the flip side of this disk, runs from the menu program.

GEOS

Steve Vander Ark

New Products

Tonight, on GEnie in the Commodore Flagship area, I read an announcement that Maurice Randall is unveiling a brand new program for the Commodore this month. The program is a Commodore fax package, just one more example of the versatility and power of the Commodore 8-bit computers.

In that same announcement was mention of another exciting new product, an 80-column super-application for GEOS called *Finally*. Maurice has been working on *Finally* for several years now, and when it actually hits the streets, it might very well be the most powerful desktop publishing program the Commodore has ever seen.

Big things are happening! GEOS is still alive and kicking and showing its teeth. There is online support available on GEnie, which is as close as a modem phone call, and Creative Micro Designs is always coming out with bigger and better products. And right smack dab in the middle of it all, *Gazette* is coming to an end.

This is my last column for this magazine. It was only a matter of time, I suppose, since the parent magazine, *COMPUTE*, was bought out a few months ago, but that doesn't make it any less sad to see it happen. After all, I still have plenty of things to write about!

So, rather than spend my last thousand words crying in my coffee, I'll spend it telling you about a few more tidbits I've turned up in the last month or so.

For one thing, Maurice Randall's *Finally* desktop publishing program is supposed to include a full-power version of his *geoMorph* program. *geoMorph* gives you the tools to take one photo

scrap image and morph it into another one. The results are very slick. Not *Terminator 2* effects, maybe, but pretty darned impressive nonetheless.

If you want to try the current, demo-powered version of *geoMorph*, check out the Flagship on GENie. You can keep up on all of Maurice's programming adventures there, since he is a regular visitor to that area.

Another project Maurice has been working on is the upgraded native-mode drivers for *gateWay*. Native mode partitions are a type of partition on a RAMLink or hard drive that can be a variable size, that do not emulate a particular disk drive type. Maurice has been struggling with some very insidious bugs in the existing drivers, but he said recently that he had almost squashed them all. Maurice is a very thorough guy, so you won't see these drivers released until they're perfect. Again, check GENie out for current updates on this.

He even suggested once that he might make native-mode drivers for the standard *deskTop*, which would really be exciting for us icon-lovers. If you use *DualTop* or *geoSHELL*, never fear. When you launch those programs from *gateWay*, they'll automatically use the native mode drivers and be able to access the larger partitions as well.

Maurice Randall isn't the only GEOS programmer who has been keeping busy. Jean Major has reportedly come out with a new and improved version of his classic patch *dbGetFiles*. That program modifies any GEOS file requestor dialog box to allow smooth scrolling of the names and to allow for more than 16 names on a list. The updated version will let you select a name from the list by simply double-clicking it. You'd be surprised how much quicker and easier it is that way!

Yet another programmer turning out interesting GEOS programs is Steve Eyrse. Steve has a knack for writing little utilities that adjust the way GEOS does things. Along the way, he manages to change some of those quirks that can drive you nuts, and give you control over things which you never thought you'd be able to control.

Let me give you an example of what I mean. Steve recently uploaded a GEOS program called SETPOINTER. (File number 16647 on GENie.) With this little utility you put the pointer wherever you want when GEOS goes into the *deskTop*, instead of having the software always place it on "view." Maybe you can't imagine why you'd want to do this. But think about it: the "view" menu is most likely the last one you would actually need to access. Wouldn't it make more sense to have the pointer start out, say, on the "file" or "geos" menu?

Steve has also uploaded a patch program for the *Maverick* boot disk creator. This patch removes the "master disk" status from a boot disk created by *Maverick*, which means you will be able to copy and erase files much more easily. You never know what Steve will come up with next, but you can bet it will be a neat program to have.

But how will you find out about it? I'm afraid it won't be here, but there are other magazines still around. And there is no source so immediate or as interactive as GENie. There are plenty of experts around there to answer your questions. Every evening you can find other Commodore users in the conference rooms. And when Maurice Randall or Steve Eyrse have something new and exciting to share, they upload it to the Flagship.

You can browse through libraries of GEOS files to find everything from fonts to clip art to full-fledged applications. Here's how you can sign up.

1. Set your terminal program for half-duplex (local echo) at 2400 baud.
2. With your modem, dial toll-free in the US (800)/638-8369 or in Canada (800)/387-8330.
3. Upon connection type HHH
4. At the U#= prompt, type JOINGENIE and press Return .
5. Have a major credit card handy, or your check book if you live in the US. You can have the charges automatically deducted from your checking account for an additional \$2.00 monthly fee.
6. Once you're on line, type COMMODORE at any prompt to get to the Flagship.

This is it for my GEOS column, but I certainly hope to see all of you on GENie! When you get there, leave me some email and say hello. My address there is S.VANDERARK. (If you're an Internet user, by the way, you can send me email using this address: s.vanderark@genie.geis.com.) See you later!

PD Picks

Steve Vander Ark

It happened. I knew it would. I even said so in at least one of my columns. Now the time has come, and I don't like it one bit.

I'm not talking about this being the last "PD Picks" column I'll ever write, I'm talking about placing a program here in this column and on our *Gazette* disk that isn't really shareware or public domain.

It finally happened. A few months ago I wrote about a great air hockey game called *Face Off*. I raved about the great game play, about how fast the puck zipped around the table. As far as I knew, this was a public domain game. I couldn't find any evidence to the contrary. So I wrote it up and put it on the disk for you to try out.

As it turned out, *Face Off* was in fact a copyrighted program. It was written by Kevin Mykytyn and Mark Tuttle and it appeared in (get this) the January, 1985, issue of *Compute's Gazette* magazine. Yep, it's one of ours, and Kevin and Mark were *COMPUTE* programmers. Ron Tulppo, a reader from Clearwater, Florida, sent me a nice letter informing me of this fact and even enclosing a photocopy of the article itself. (*Good catch. It's relatively easy to find those references with the modern internet, but wasn't so easy in 1995. -MS*)

Well, I have to admit I am a little relieved that the copyright on this program is actually owned by our own magazine. Please note, of course, that this program IS copyrighted, and even though this column incorrectly implied otherwise, it is not available for users to copy freely and give to friends. All rights remain with Compute Publications. This applies, by the way, even after this month's final issue goes to disk. It applies even though *COMPUTE* is no longer published. This program, along with all the other programs and articles from all those years of magazines, is still firmly copyrighted and will be for quite a few years to come.

But hey, I don't want to leave you all with a fire and brimstone sermon about copyright laws. I want to leave you the way I came in with this column: kicking up my heels and tearing through a selection of great Commodore programs. So enjoy this month's selections.

Cosmos Designs

Hold on to your hats. try this baby. It's what you call a "demo," which means it's just some hacker's oddball combination of music and graphics designed to show off his programming ability. In this case, though, the hacker involved is one talented fellow!



For starters, the music is absolutely top notch. You'll swear there are more than three voices playing. For another thing, the high-resolution graphics in the background are superbly drawn, with excellent three-dimensional shading.

But the animation is what really sets this demo apart. Most demos use two kinds: scrolling text and moving sprites. The form their animations take is really the true measure of a demo

programmer's art. And let me tell you, these scrolls and these sprites are incredible. There are falling globes which explode into sparkling fragments and a head that bounces off the scrolling text as if the line of letters were set in elastic. You really have to see this to believe it.

Warriors

I've saved the best for last—really last, I'm afraid. This program is very large (197 blocks all told) and very well done. The graphics are exquisite and the animation right on the money. You are a warrior whose mission is to charge across a landscape defended by a variety of nasty soldiers with the goal of killing the king and taking his crown. Along the way you hurl an endless supply of battle axes at everyone and everything. You can play alone or with someone else racing with you.

Ok, so maybe it isn't exactly politically correct to play games filled with mindless mayhem and regicide, but it could be worse. The overhead point of view is beautifully rendered and while death constantly surrounds you, there is no blood and only a sort of mystical shimmer when you catch a knife in the midsection yourself. Hey, you can make up some justifying back story for yourself it will make you feel less guilty about hacking up a king. Deal with it; this game is a rush. Instructions are included with the program.



That's it, folks. Enjoy the programs!

Hints from a Heavy User

Don Radler

From organizing software to backing up disks, here are some tips to make your computing more enjoyable.

You could call me a heavy user of the grand old Commodore 64. I do writing on it, some cartooning and some music, and I publish a modest line of my own software.

The software business calls for much BASIC programming, debugging and compiling, and ultimately, a lot of disk copying. It also calls for a bit of desktop publishing and some corresponding, plus recordkeeping with a database, and bookkeeping with a spreadsheet. (I also play an occasional game and take a simulated flight now and then.)

Thanks to all this activity, I keep on discovering new uses for this supposedly obsolete 8-bit machine, and I keep learning new ways to exploit it. I first encountered the 64 in the mid-'80s, and started teaching myself how to use it. Almost from the outset, *Compute's Gazette* was my only tutorial. Back then, the name of the magazine still had the "!" implausibly perched ahead of the apostrophe, but it was already the best magazine in the field. I'm still helped by it all the time.

I'd like to give something back to *Gazette* and its readers by passing along some tips from my own long voyage of discovery with the 64. I'll try to distill the essence of it in this article.

Software Is First

Although it may not seem that way when you first lay out money for a computer, software is more important than hardware. Unless you buy your machine for one single purpose, or give up on it right away and stash it in the attic, you'll put much more money into software than into hardware. And much more investment of your time organizing and using that software, even if you don't do much programming. So I'll start with software, and try to arrange things in their order of importance.

Back Up Everything

There is nothing more important than backing up your disks and your individual programs. Some copy-protected disks can't be backed up easily; for any really important disk, buy a second copy from the publisher (almost always at a substantial discount). For other disks, get a good disk copier and back up everything.

In addition, get a file copier (many have been published in *Gazette*) and again, back up everything. Backup even partial programs as you write them. You'll fill up a lot of disks and you'll surely reformat many of them to kill old stuff and make room for the new, but you'll never regret having made backup copies in the first place. There's nothing like the sinking feeling that comes from accidentally corrupting a disk directory, unless it's the offsetting feeling of triumph when you find the backup and know that nothing is really lost.

If you're like most Commodore users, you have one 1541 drive. In the beginning, that's the only drive there was. And even today, with new releases a rarity, most software is written for the 1541. Since most users work with a single 1541, I'll confine my suggestions to that setup. If you have two 1541's, or any other drive, you will, of course, modify these ideas to fit your system.

You're probably used to working with single-sided disks, because that's the way most software comes. But floppies are fine for backups and they save money in addition to saving space. As Calvin Guild pointed out in *Gazette* way back in March 1993, there's really no technical reason not to use these two-sided disks with a 1541 drive, whether you buy them this way or use a disk-notcher to make your own. And if they further your habit of making backups, they're a net plus.

Organizing Software

Develop your own method for organizing and filing your disks. The time you save as a result will be significant, and the ease and pleasure with which you use your entire system will be enhanced immensely. Store your disk collection in file boxes that can separate categories with movable partitions.

Here are some possible categories:

Current Work, Utilities, Databases
Word Processing, Desktop Publishing
Graphics, Sound, Science, Languages
and Games, of course.

You'll probably prefer a different breakdown, or use fewer or more categories. And at some point, you'll probably need more than one box - with software, as with potato chips, you can't stop with just one.

Buy disk labels with assorted colors and key them with four-letter codes like UEQP for equipment-testing utilities and DCAT for your disk catalog. Such codes are better mnemonics than are the less meaningful two-letter IDs. You can also spell out the disk title on the label, and you can add notes that are significant for loading, such as DFL for Disable Fast Load or XFL for remove it completely.

Put your disk catalog program on side A of a floppy, with the files on side B. That way, you're working with one disk instead of two. When you update a file, you can write it to side A, then scratch the one on the flip side, copy the new one from side A to side B, and then erase it from side A. This will work with any kind of database program, of course.

Arranging Directories

To make individual programs on a disk more convenient to find, alphabetizing helps. I used to do that with a program from the March 1985 *Gazette*, compiled for speed. Recently I switched to a newer and more powerful utility by Tracy Eichheim from the October 1987 issue. *Directory Magic*, as it's called, will alphabetize for you. But it will also let you move any listing within a directory to any position, scratch and unscratch listings, move scratched titles to the top or the bottom of the directory, and then validate the disk with its new directory. It's a topflight tool for directory management.

Loading Programs

All 1541 users should try a cartridge such as Epyx's Fast Load cartridge to speed up loading. Fast Load lets you load the first program on a disk by pressing the Commodore and Run/Stop keys simultaneously. Then, on disks with more than one program, put the Fast

Load version of Art Hunkins' *Keyload* from the July 1986 *Gazette* as the first listing. After it loads and runs, simply cursor down to the program listing you want, hit Return, and that program is up and running.

There are more powerful cartridges than *Fast Load*, some of which also speed up saving programs and perform other chores like screen capture, but Epyx's old, familiar and inexpensive workhorse will serve you very well indeed.

Many software manufacturers use copy protection schemes that cause the head on a disk drive to knock. This does the drive no good at all, and ultimately can lead to alignment problems. To prevent this, try a routine like this one I picked up somewhere along the line:

```
10 OPEN15,8,15
20 PRINT#15,"M-W";CHR$(106)CHR$(0) CHR$(1)CHR$(133)
30 CLOSE15
```

The spreadsheet I use has a head-knocker built in, so I put the above little program first on the flip side, where my spreadsheet files are kept. A quick press of the Commodore and Run/Stop keys installs it, and then I turn the disk over to side A and load the spreadsheet with another keypress - and without any head knocking to send my drive out of alignment.

Utilities

Back in the September 1992 *Gazette*, I described the basic utilities on my work disk at that time. They've changed some, but not enough to warrant further discussion here. Three things bear mention, however.

If you do use *Fast Load* and also *Blitz!* as your compiler, you know that they are incompatible. Try putting Broderbund's *KwikLoad* on the disk along with *Blitz!*, and a note to yourself on the label to disable *Fast Load* and install *Kwikload-1* to load the compiler (DFL/KW1).

You can also remind yourself of such requirements right in your directory, too. Since the Commodore saves a shifted space as a quote mark, if you save *Blitz* as BLITZ-Shifted-Space-KW1 it will appear in your directory listing as "BLITZ"KW1. The note after the second quote mark won't prevent normal loading with just the name BLITZ or with B*.

If you compile at all, it's a good idea to maintain a disk on which you store the original listings, in BASIC, from which you compiled the program that you end up using. That way, if an afterthought comes along, you can go back and make changes and then recompile the edited program. UOLI seems a natural mnemonic for the original listings.

If you call machine language routines from the BASIC programs you write, try storing them on two separate disks, one for those that start at the much-used 49152 address, and one for those that start elsewhere. That way, you won't have to try to remember whether a given routine is compatible with the program you're working on.

Finally, you might want to consider using a RAM disk, such as the one by Hubert Cross in the *Gazette 1988 Special Issue*, for disk copying chores. With a program such as *Directory Magic* to clean up directories, another like Lou Sander's *Disk Name/ID Changer*, and a disk copier such as the Krackerjax *C64 Fast Copy* all in 64 RAMdisk, you can clean up a bunch of disks and finally make copies of each one without constantly loading utilities as you work.

Word Processing

If you use a small word processor such as *SpeedScript*, you can place it as the first program on a number of disks, with the files it produces right on the same disk (and on the flip side if needed.) The version of *SpeedScript* 3.2 that I use has the 80-column preview and Easy Cursor features added, along with the elimination of the Tape or Disk? question.

The bug-prone save-with-replace routine has been eliminated, too - instead, the SAVE:@: command scratches first and then saves. This version has never messed up a save.

Right after the WP program (which I've named 0SS/80/EC) comes 1CUSTOM SETTINGS with the codes for draft mode, italics and underline and so on, and 2HEADER with a one-line template for that amenity. Either or both of these can be loaded before you start to produce a document of any kind, or, if you forget to do that, they can be loaded at the end of the document, erased, and then restored at the very top. I temporarily name the article or other document I'm writing 3ARTICLE. That way, you don't even have to remember the title: just load 3* once your word processor is installed and you're back in action. Replace and save the new versions as you develop them.

Try putting LTRS on one WP disk, ARTC on another, and MISC on another, or whatever categories suit your word processing needs. Keeping them separated this way may use more disks, or disk sides, but it will speed up all your work.

Auxiliary programs for word processing, such as *SpeedScript's File Convert*, *File Search*, or *SpeedSwap* can all go on a separate disk in your WP section. USSX is the label I use for such extras. Often-used auxiliary programs, such as WORD COUNT, can go right on the disk where they're employed - ARTC, in my own WP setup.

Desktop Publishing

You've read reviews of desktop publishing (DTP) programs in *Gazette*, and I won't duplicate any of that. Having tried several DTP programs, I settled on *The Newsroom* from Springboard Software. I use it to produce the flyer for my own line of software and it does a fast and totally satisfactory job. Still available by mail order from various sources, the program and its several clip art disks are a real bargain. Just by following the well-written manual, you can produce newsletters and other documents, as many user's groups have done and as many grade school and high school classes have also done while learning the rudiments of publishing.

Right up front, the manual tells you that you can use either the keyboard or a joystick to put the program through its paces. This may be the factor that sold me on *The Newsroom*—I just don't like working with a joystick or even a mouse. Most of *The Newsroom's* graphics devices, such as the box routine, actually work faster with the keyboard than they do with a joystick. And the zoom routine that lets you alter individual pixels is quick and precise using the keyboard.

The manual fails to mention some possibilities, however, that might be of interest to you. For example, what *The Newsroom* calls a "photo" prints in the center of a page, while a "panel" prints flush left. You may think of ways this can serve you. I used it to draw cartoons in "photo" format, and sold a couple to RUN Magazine before it joined Ahoy! and Commodore Magazine in 8-bit heaven. (Unfortunately, RUN's untimely demise preceded any publication

of those masterworks.) The Newsroom manual does mention that you can print more than one “banner” on a page—you might find interesting ways to exploit that.

For producing a printed document, don’t think you’re limited to using any one program. You could, for example, start with a banner from *The Newsroom*, add some big type lines from a program like John Robinson’s *Poster Printer* in the November 1987 *Gazette*, then go on to some 80-column text from your word processor, add vertical bar charts from a program like Tim Ruiz’ *EZ Bar Charter* in the September 1989 *Gazette*, or add hi-res art created with a program like *Doodle* or *Blazing Paddles*.

If your printer supports it, you could even print the art in color—there’s no rule that stops you from changing ribbons in midstream. The point is, the Commodore is a flexible computer and there are thousands of good programs for it kicking around; the end product is limited only by your ingenuity as a Commodore user.

Hardware Hints

My own hardware setup is bare bones indeed—a 64, one 1541 disk drive with a *Fast Load* cartridge to goose it along, an 1802 monitor, and an Okimate 20 color printer. Most of you reading this probably have a more elaborate setup, and are looking forward to further upgrades.

The simple system I have has been in almost constant use for something approaching a decade. I’ve replaced the power supply and had the printer tuned up once; everything else just keeps grinding along. I’m inclined to believe that proper installation and maintenance have contributed to that remarkable record.

Dangers to Avoid

The power supply builds up heat, and seems to suffer from some sort of death wish. Most computer failure stems from power supply problems, so this unromantic box is something to be reckoned with.

Keep it off the floor, or off a shelf, so that air flows all around it at all times. When that precaution proves insufficient, replace the thing with one that is fused and repairable. After doing that several years ago, I haven’t had to replace the fuse, much less repair the box, but I’m glad that I can do either if I need to.

Plug the power supply and the rest of your system into a surge suppressor with an on/off switch; control everything with that switch. Years ago, in all the magazines, there were endless discussions about the order in which things should be turned on. Today, you don’t read about that at all. In my experience, one switch can control everything with absolutely no problem.

But don’t believe that any surge suppressor can protect your system against a lightning strike. Where I live in south Florida, these are endemic, and I’ve learned to simply unplug the main line when storm warnings are issued.

Heat, dust, grease, and dirt are enemies of all equipment. You know that as well as I, so that’s all I’ll say about them. I have found that the upholstery-cleaning brush on a vacuum cleaner works well on everything from the keyboard to the monitor screen, and suggest it as a

cleaning device. Liquid cleaners and such are unnecessary and could be downright prejudicial to the health of your equipment.

Disk Drive Alignment

The next most frequent hardware problem is with disk drive alignment. I suspect that head-knocking from copy protection schemes causes some of the problem. The program I said I use with my spreadsheet also works with some commercial games and such, but not all—try it with your copy-protected software and use it where it works.

I check alignment and speed regularly with *1541 Physical Exam*; they've held up after millions of revolutions. Part of the reason may be regular cleanings with the disk spinning for a full 30 seconds. As Calvin Guild pointed out in his article cited above, over-cleaning can be harmful; four times a year seems to have done it for me.

To keep your drive spinning for about 30 seconds, try this program:

```
10 OPEN15,8,15:OPEN2,8,2,"#"
20 PRINT#15,"M-W"CHR$(106)CHR$(0) CHR$(1)CHR$(133)
30 FORI=1TO65
40 PRINT#15,"U1:"2;0;Y;0
50 Y=Y+1:IFY>35THENY=1
60 NEXT
70 CLOSE2:CLOSE15
```

Finally, a disk drive also generates heat and needs to have air move around it. If your desk doesn't close in the back behind the drive, and there's air all around it, there should be no problem.

A Printer Tip

I have one final tip for other users of thermal printers like my Okimate 20: use fax paper. With no ribbon at all, this paper will give you much better impressions than you can get on extra smooth thermal transfer stock using a ribbon, and at much lower cost. With everyone accustomed to getting fax messages, this paper will serve for all but the most formal work. And it will give you a sharper master if you're going to photocopy from it.

The only real drawback is that the paper comes in continuous rolls and isn't perforated, so you have to roughly measure pages as you tear them out of the printer and then trim them to clean up the top and bottom edges.

(Writing this, I wondered if someone would ask about using fax paper with a ribbon, black or color, so I tried color on one of the hi-res pictures from a *Gazette* disk. Forget it.)

That's it. Those are some tips that I've discovered over the years to help me get the most out of my Commodore system. Like most things in life, if you put a little effort into it, you'll be rewarded. And I'm sure if you put your mind to it, you can come up with your own ways to get many more years of productive life out of this supposedly obsolete 8-bit machine.

Steel Trap

Larry Cotton

Want to have a memory like a steel trap? Don't waste a lot of time and effort trying to develop one, just use your computer's.

"Don't you remember? I discussed that with you at some length on February 24. You agreed to send it to me immediately. What happened?"

The above scrap of conversation is one that transpired between a supplier of parts for the company I work for and myself. I know because I take good notes.

How? With my *Steel Trap*—my unbeatable combination of a computer and a word processor.

I've never been gifted with the ability to remember things long enough to forget them. This is especially true for the things people tell me, or swear they'll do for me. Worse, I used to forget to do things that I had agreed to do. My Responsibility Quotient was taking a nosedive as my aging memory cells gradually deteriorated; my RAM seemed to be at capacity.

Before discovering the Steel Trap, I'd contracted a severe case of the Post-It syndrome; I was a slave to a multitude of memos. My desks at home and work were little more than horizontal surfaces gradually sagging under the weight of millions (OK, hundreds) of tiny scraps of paper with telephone numbers, airline schedules, appointments, and even the occasional recipe for beer bread. Of course, the notes I really needed became buried under a yellow blizzard of new notes, and I would ultimately forget to do something or call somebody.

I bet this doesn't sound at all familiar to you. You're probably super-organized, have one of those Day-Timers, and all's right with the world. You're probably remembering to send flowers for your wife's birthday at this very moment.

But if by perchance you're more like me in the organization department, be prepared to become suddenly more responsible, trustworthy, and much more efficient. You'll amaze your family, friends and co-workers with your newly energized powers of recall. Your vocabulary could even expand. And all this without the first "How To Improve Your Memory In Only 30 Seconds" course.

The 'Ware

You wouldn't be reading this if you didn't own a computer. The type of computer's not particularly important; a 64 or 128 will suffice nicely for your home or small business.

If you own one of these computers, odds are that you also have a worthy word-processing program, such as *Speedscript* (written by Charles Brannon, formerly a *COMPUTE* editor). You have all the trappings necessary for the Steel Trap.

Speedscript has served me very well in the home office for years. Although it isn't particularly sophisticated compared to the word processors written for the PC clones, it does have the key features you'll need: editing and searching. More on those in a moment.

The word processor I use most at Bosch Power Tools, my place of employment, is *Microsoft Works*, version 3.0, for DOS. (It's actually an integrated program, consisting of a simple word

processor, rudimentary spreadsheet, database and communications software.) Any word processor with easy-to-use editing functions will help you create your Steel Trap. I'll describe how I use *Speedscript*; the principles are adaptable to any word processor.

Setting Up

First, make sure your phone and computer are next-door neighbors, and that any pads of paper are hidden, or at least more difficult to get to than your computer. Now, run your word processor and save an empty file (think of it as a document with nothing written in it yet) called STEEL. (It doesn't really matter what you call it as long as you can remember it and it is quick to type.)

STEEL is basically a to-do list. But it's far more than that, as you'll soon see. A good way to begin STEEL is to just enter a few items from the pile of paper notes on your desk. Don't number the entries. They'll very quickly get scrambled. Would the numbers serve any other purpose than to prioritize your list? We'll do that another way. Read on. Make each entry a separate paragraph—not indented, but with a blank line between each one, just like the paragraphs in this article.

Now, locate your word processor's edit keys. In *Speedscript* we'll use Ctrl-E for erase and Ctrl-R for restore. Use Ctrl-E to move an entry into the buffer, then Ctrl-R to spit it out (move it) where you want it.

Why move entries? We do it to place them in priority order: higher priorities on top; lower priorities, or accomplished tasks, at the bottom of the document. Here's an example of the life-span of a typical short-lived entry:

The phone rings. I don't grab a piece of paper and a pen; I load STEEL, if it's not already open. It's my wife, reminding me to pick up a pizza on the way home from work. Since that's a pretty high-priority item, I immediately type it in at the top of the document. Don't write it down first! Get rid of your little paper pads and the habit of writing down something to be transcribed later to the computer. Go straight to the electrons.

Keep your entries brief. In this case the word "Pizza" should be enough. As your mind deteriorates further, you can always make your entries longer, such as "Buy pizza on the way home from work, dummy".

Check your list before leave the office. After work, you pick up a four-topping pizza at your favorite Italian bistro. Your wife is amazed. The next day, fire up your word processor, open STEEL, and delete the pizza entry.

Telephone Numbers

Here's one way STEEL is more than just a to-do list. I recommend that you include a few important, frequently used telephone numbers as stand-alone entries. With the numbers, add a person's name, maybe a company name, and possibly a remark or two regarding that person or company. If there's no particular activity going on at the moment with that person or company, a typical entry could be:

Sammy (talks a lot) Johnson; Reynolds Enterprises; 42 E. West St.; 919-123-4567

Try to limit the phone numbers to those which are relevant to particular projects, or that you call frequently. I would not recommend mixing your entire phone database with your to-do list.

Telephone Notes

Open STEEL every time you make a phone call, and every time someone calls you. If you need to call somebody soon, just enter his or her initials near the top of the list. If that person's already in your to-do list with a phone number, move that entry to the top of the list. If you must call someone at a particular time or for a particular reason, add that information.

While you're chatting with that person, you may want to add a few notes to the entry. If the conversation involves something for you to do, leave the entry near the top of your list. If not, note the date you placed the call and move the entry to the bottom of the list. After you've hung up, delete any superfluous information.

(Incidentally, I love the quietness of the 64's keyboard. My computer at work is much noisier. A person on the other end of the phone line may hear it and think you're going about your business while they babble on, but *au contraire!* You're actually taking notes about the conversation.)

Status of Projects

My 8-to-5 job entails tracking the status of myriad projects—some big, some small. Say your boss asks you to begin the task of introducing a new power tool to the world. Say it's a cordless drill. Also say that the deadline for the drill's introduction is May 5, 1996. Your new paragraph, which begins life near the top of the document, could be:

2/7 (today's date); cordless drill; intro deadline 5/5/96

That's it! As you do things to bring the drill to market, add them—with dates—to the growing cordless drill entry. I make it a habit to limit each entry's length to one paragraph, and to separate thoughts with semicolons. Be sure to include key words, such as "cordless" and "drill", which you can search for later. Three days hence, the next thought could be:

2/10 talked to Sandra about purchasing options; she will get back to me by 2/17

Aha! There's your first tickler. If you don't hear from Sandra as promised, on 2/18 give her a jingle. She'll be amazed. Daily, as you peruse your rapidly expanding STEEL file, add to the paragraph; when things do get accomplished, note these as well—with the dates!

If the cordless drill entry becomes too unwieldy, I will create a separate file to help manage that project.

Personal Stuff

My to-do list at work is 99 percent business; at home it's 99 percent personal. I would not mix the two, unless you have a very understanding boss, or you are the boss. But even in a business to-do list, an occasional personal, short-lived entry such as "Buy pizza" or "Call TH" would be permissible.

My Steel Trap at home contains, among other things: where I saw magazine articles, my beer-bread recipe, basketball game schedules, little-known facts (1 mph = 2688 furlongs/fortnight) airline ticket prices, vocabulary builders (my favorite: “Planomania: a pathological desire to wander and to be free of obligations”), and my sons’ temporary address and phone number in California. I also have countless projects such as fix garage door, sell washer, check Honda brake fluid, attach dryer hose, and so on. I’d be really surprised if you don’t have a to-do list or a job jar, so why not computerize it?

I recently fenced in our back yard. In my Steel Trap was one big paragraph on costs, hardware store phone number, things to buy (such as pickets, hinges, latches, corner braces, lead anchors), results of calculations on how many pickets would be required, and so forth.

Ideas

Everybody has ideas. Idea generation may be part of your work, or a random thought that you may need to remember may simply pop into your head. Don’t write it down—put it in your Steel Trap. Each new idea or thought should probably be a stand-alone entry. Later as you elaborate on your ideas, expand the entries.

Occasionally, browse through the entire document. An idea may be ripe for use or may inspire further thought. Conversely, what seemed like a brilliant idea at 4 a.m. may look like a real dog in harsh daylight, so just delete it.

Hunting

Eventually, your to-do list will span several pages. (I try to limit mine to 10.) But even with a multipage list, it’s quite easy to find entries; just use the word processor’s built-in Hunt or Search function.

Say your boss wanders into your office and asks for a status report of the cordless drill. Open up STEEL and execute a search. In *Speedscript*, press Shift-Ctrl-H and enter the word or phrase you’re looking for, such as “cordless.” Ctrl-H then finds all occurrences of that word. Use a word that you know is in, and unique to, that entry. (In *Speedscript*, you must also ensure that the case of the letters as well as the spelling of the word match.)

Almost instantaneously, your status report is there for the world to see, complete with dates milestones were accomplished, phone calls you made, promises others made, and so forth. Let your boss read it, preferably on-screen. He’ll be amazed.

You can also use Hunt to find your often-dialed telephone numbers. Just type in “Sam” or “Sammy” and hit return.

Moving and Deleting

So when do you move entries and when do you delete them? Move entries to the bottom of the list as their priorities lessen, or after a job’s been done. That will let others bubble to the top. Leave the accomplished tasks at the end for a month or so, then delete them after the following has occurred:

- You've done what you're supposed to;
- Everybody else has done what they're supposed to (if anyone else is involved);
- Enough time has passed that you won't need to prove tracking;
- You've made a hard copy of the entry and filed it with any other relevant papers.

Saving

Save STEEL often. With the 64 or 128, you can either give the document a new name, such as STEEL 2-7-95 or (in *Speedscript*) use Ctrl-^ (up arrow) and S to scratch the old file before you save it with the same name.

Printing

STEEL rarely needs to be printed in its entirety. Occasionally, you may want to print out a single page which contains a status report for the boss, or after a project has been seen to a satisfactory conclusion, or even to take with you to that hardware store.

Make STEEL an important part of your life. Before you shut down your computer for the day, scan it one more time. If you do that, the Steel Trap won't let you forget the pizza. And even you will be amazed.

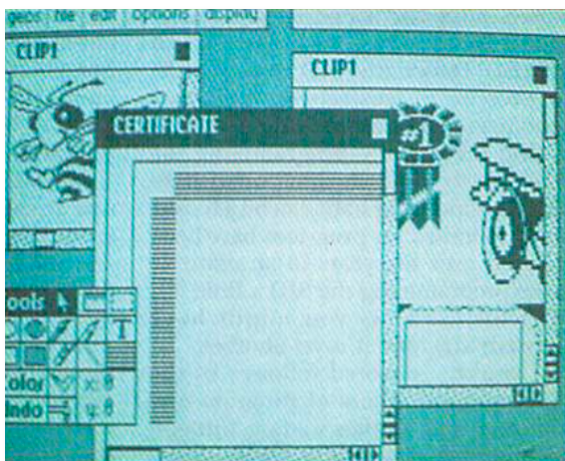
Review: geoCanvas 3.0

Steve Vander Ark

See what's new with Nate Fiedler's alternative graphics program for use with GEOS.

I love creating computer artwork. Naturally, I do a lot of my graphics work in GEOS. So when I received a copy of the latest version of *geoCanvas*, I was delighted.

Nate Fiedler had done an impressive job on the last version of his graphics program for GEOS, and the new version promised to be even more spectacular. For starters, it was a 128 product, designed for the 80-column screen. On top of that, Nate had redesigned the way the program used tools. I couldn't wait to give it a try.



Before I charge ahead into a description of some of the program's new features, I had better give a brief introduction to *geoCanvas* for those of you who may not have heard of it before. *geoCanvas* is an alternative graphics program for use with GEOS. It uses many of the same tools and features that *geoPaint* has, although each program has its own strengths and weaknesses.

One of the most impressive features of *geoCanvas* is its ability to display more than one window at a time, each one showing different section of the same or different documents.

Of all the tools in *geoCanvas*, the one which really surpasses *geoPaint* is the text tool. *geoCanvas* lets you change fonts and styles within your text block, which makes it possible to do rudimentary desktop publishing right in the graphics program.

Version 2.0, which is strictly a 40-column product, was released by Creative Micro Designs some time ago and was an instant success. That version of *geoCanvas* as well as the new version are now being distributed by Nate himself. His address is at the end of this review.

geoCanvas v3.0 is an exciting program. All the great features of version 2.0 are still there, including multiple resizable drawing windows and excellent text handling. On top of that, more and more features had been added. These features are activated from a new menu bar item called “tool attr.” There you can set anything from “constrain” on lines and shapes to such nifty drawing features as rays.

When you load a new tool, the contents of this drop-down menu change to show all the options for that tool. And what a selection of options! Nate has thrown in just about everything you could imagine.

The tools themselves are now separate files on the *geoCanvas* disk. You can load into your working copy of *geoCanvas* whichever tools you think you’ll use. Since each tool is a separate entity, you can also replace a single tool with an upgrade or add more tools should they become available. All you do is replace the tool file itself with the new one and that’s what *geoCanvas* will use. While this does clutter up the directory a little, this is hardly a problem and certainly a very small price to pay for the flexibility it offers.

geoCanvas supports several video modes. Depending on whether or not your 128 has additional video RAM, you can use 200 or 176 screen lines. In some of these modes you can include color, something *geoPaint* users have always missed in 80 columns. Of course, the image on your screen is still going to appear squashed horizontally; that’s inherent with the 80-column screen on the Commodore. For that reason, it would have been nice to have the choice to drop into 40-column mode once in a while. However, the 128 *geoCanvas* 3.0 is just as firmly locked in 80 columns as version 2.0 was in 40. The preview option shows your page with the correct aspect ratio, though, so you can always keep tabs on how things are looking.

All this power requires a RAM device to operate, and if you aren’t careful some programs could find themselves fighting with *geoCanvas* for space. I found, for example, that I had to load *geoWizard* after *geoCanvas*; once installed, however, the two programs work great together. As a matter of fact, I’m switching between *geoCanvas* and *geoWrite* as I write this review, using *geoWizard*’s task switcher.

I really like this program. It’s not perfect, that’s true—the way some tools work isn’t always very intuitive, for example—but Nate has really gone the extra mile to make things user-friendly. There are plenty of keyboard shortcuts to cut down on the menu-hopping. The cursor changes into a miniature version of each tool icon so you know which one you have active; it also automatically changes to the pointer tool when you want to adjust the window. I suppose I could nitpick about liking some little feature or other better in *geoPaint*, but hey, I can always open a *geoCanvas* file right from *geoPaint* and use those tools on my graphics too, if I want. The two programs together make an extremely powerful set of graphics tools.

On the disk with *geoCanvas* are several other useful programs, the most exciting of which is *ScrapCan*. This utility allows you to cut photo scraps out of a Paint or Canvas document of any size, then paste those scraps wherever you want on that same or another *geoPaint* or *geoCanvas* page. ScrapCan is very intuitive, since both cutting/copying and pasting are done on preview-style representations of the page.

geoCanvas is a very capable program. It makes good use of the latest in Commodore technology, including extra video RAM and the current crop of high-end RAM devices. The feature list is very impressive and the whole thing is about as user-friendly as so complicated a program could be. I recommend *geoCanvas* version 3.0 very highly.

geoCanvas

Nate Fiedler

5711 Mt. Pleasant Rd.

Bernville, PA 19506

\$28.00

Programs

Satellite Tracker

Donald G. Klich

In order to point your dish antenna at the proper satellite, you have to know the satellite's coordinates. This program plots more than 50 of them for you.

If you have been giving some thought to purchasing a satellite dish television receiving system, here is a program that you should use before getting too involved. *SatTrack* will plot all of the current communication satellites in geosynchronous orbit. Included are special service and television satellites in both the popular C-Band and K-Band.

Before purchasing a satellite TV system, it is important to determine if your site has sufficient line-of-sight capability to reach all of the desired satellites. If you live in a restricted city area, perhaps an economical system aimed at a single satellite is sufficient. Others in a more open environment may wish to have a system that can track many satellites. Before determining which satellites are visible to you, and where they are located, you must first find out exactly where on earth you are located.

Your exact longitude and latitude can be found on quality maps that should be available at either your local public library or city hall. *SatTrack* requires input in degrees, minutes, and seconds. Also required is whether your longitude reading is East or West of the Greenwich zero meridian. From the same sources you should be able to find out how many degrees off true North, the magnetic North is for your location. The examples included in the program happen to be my coordinates in Illinois.

After entering your site information, the program checks a file of almost 100 satellites and lists all that could be visible to you. The two items needed to locate a satellite are its azimuth and elevation. The azimuth is the angle from True North (0 Degrees) you must face; East is 90 degrees, South is 180 degrees, and West is 270 degrees. Elevation indicates the angle your dish must be tilted upward; 0 degrees being the horizon and 90 degrees being straight up.

Using these two pieces of information, it is possible to determine if a clear view can be obtained by using a compass and some form of an adjustable triangle or protractor. If buildings or dense trees block your view, reception from that particular satellite will be unsatisfactory for that site location. Moving to a different spot may correct this problem.

Library reference books can be consulted to determine which satellites transmit information of interest to you. *SatTrack* is designed to work for any point on earth, and as a result you must allow it to process even those satellites that are below your horizon in case it finds additional candidates as it completes its cycle around the earth.

SatTrack was originally written in BASIC 7.0 for the 128 but a slightly slower version for the 64 is also included on this disk. The data statements at the end of the listing contain valid 1993 information. Eventually this list will need to be updated as satellites are replaced or added.

Screensaver

Martin Fensome

Screensavers are meant to be used by BASIC or ML programmers to avoid monitor burn-in while programming. They are also useful for anyone who leaves the computer unattended for any length of time.

These screensavers should work with BASIC and ML programs that do not:

- (a) Overwrite them in memory.
- (b) Change or disable the interrupts.
- (c) Disable the keyboard.
- (d) Do anything else that I didn't think of.

There are three versions on the flip side of this disk: a 64 BASIC version, a 64 ML version, and a 128 ML version.

Version 1 - 64saver

When you run the BASIC version, the ML code is contained in data statements. The title and copyright information is printed and the code is poked into memory. A message informing the user that the Screensaver is installed is printed along with the enabling and disabling SYS commands.

The user need only enter SYS 52970 to activate Screensaver and SYS 53077 to deactivate it.

If you press Run/Stop-Restore, you will have to re-enter the appropriate system command.

Version 2 - 64saver.52600

This version is pure ML code and must be loaded with ,DV,1, where DV is the device number of your storage unit.

SYS 52600 activates the program.

SYS 52728 de-activates it.

After the program starts, the titles and copyrights are printed, as well as the enabling and disabling SYS commands. The program then prompts the user for the number of minutes between one and nine to delay before blanking the screen. After this entry is made the program reminds the user to type NEW and press Return. Of course, if they have already entered 'NEW' there is no need to do so again.

Version 3 - 128saver6627

BLOAD"128SAVER.6627" from the disk drive which has a device number of eight. Or LOAD"128SAVER.6627",dv,1 where dv is the device number other than eight.

Either way press Return and wait while your storage device loads the program into memory. Then enter NEW and press Return.

To activate the screensaver:

Enter SYS 6627 and press Return.

To disable the screensaver:

Enter SYS 6746 and press Return.

If it has become necessary to hit the Run/Stop-Restore key combination, you must re-enable the saver with the appropriate SYS command.

How It Works

The programs get the number of minutes from the user, then set the Time Of Day clock in CIA 2 to 0. Next the TOD alarm is set to the number of minutes entered by the user. Then the TOD alarm interrupt is enabled. After this is out of the way, the normal IRQ is redirected to the keypress detection routine and the Non Maskable Interrupt vector at \$0318 is pointed to the screen blanking routine.

Until the disabling SYS command is entered or Run/Stop-Restore is hit, the computer will jump 60 times per second to the key-press detection routine. If no key has been pressed, the program leaps out and performs its normal interrupt service routine.

If a key was pressed then the clock time is reset to 0 again, preventing the TOD alarm interrupt from happening and if the screen is blanked, it's unblanked. Then the routine exits to the normal interrupt routine.

When the timer reaches the alarm time that was set, the CIA chip generates a Non Maskable Interrupt. This interrupt is vectored through RAM at \$0318/\$0319 which is pointing to the blanking routine. The first duty of the blanking routine is to check if this NMI was generated via the CIA 2 alarm time. If it was, the procedure blanks the screen to the current border color, saves the blanked status of the screen, and exits the interrupt.

This is one of the differences between the 128 and 64 versions. The 128 would always break to the monitor upon returning from an NMI with an RTI instruction or with a JMP to the normal NMI routine. A JMP to \$FA62, which is the tail end of the normal NMI routine, had to be coded to prevent this monitor break. All other NMI's are passed on to the normal NMI routine.

Screensaver Warnings

At this time, the 128 version functions with *Gazette Disk* menu and column reader programs. The 64 version will not work during *Gazette Disk* menu or column reading activities, but will remain in memory. Once out of the menu program and back to BASIC, just use the appropriate SYS command.

I have found both 64 and 128 versions to function properly with some popular copy programs (F,M, and BCOPY). They weren't specifically designed to work with anything other than your own BASIC/ML code but it's nice that they do.

Display Maker

Tom Zdanowicz

Create up to 16 screens with this character-based graphics editor. Use it to create commercial quality graphics for games or applications.

Displaymaker is a character-based graphics editor. To see what it can do, after running the main program as explained in the following documentation, use a joystick to select LOAD FILE FROM DISK option. Then type DEMO. After the files have loaded, refer to the documentation for changing colors, editing, scrolling through the 16 screens, and so on.



Introduction

My first experience with a computer was on an old 16k PET in Junior High school. I didn't get serious about programming until I bought a 64C several years ago. I wanted to write a game fully utilizing the 64's graphics capabilities but found it a long and difficult process to create full screen colorful graphics.

To make the task easier I purchased a utilities package that contained a graphics generating program. As I started to use it I found it limiting. By having to load a separate program

to create custom character sets and another to create full screen graphics displays, the process quickly became a nightmare.

After finishing I discovered that the program only allowed eight screens to be created which limits the ability to create extensive background graphics for games or other applications. To make a long story short, I told myself I could do a better job and proceeded to write *Displaymaker*.

One final word before I list the instructions. The disk contains several data files as well as the program. These are the display data and character sets for the various work screens. I intentionally took this approach to allow easy modification of the main menu and work areas. Although it takes up more space, this format makes it easier to add more menu options and features without having to rewrite long screen drawing routines each time. However, I think the greatest feature of this program is that after graphics displays are created they are

completely transportable as are the character sets used to create them. This means they can be used in any program with enough free memory.

Instructions

To load *Displaymaker*, reset the computer then type LOAD "BOOT",8 and press Return. When the computer displays the READY prompt type RUN then press Return. After the title screen appears press any key to load the main program and data files. After a few moments you will see the main control menu.

The main control menu is on the left side of the screen and the character editor on the right. You may notice that while on the right side of the screen the cursor will slow down slightly. This is done to give the user more precise control over the editor. When the cursor crosses over to the left side it speeds up again allowing more rapid selection of options. It is not necessary to stop on a selection number before pressing the joystick button. The button can be held down as the cursor passes over the number. This is useful to quickly change display modes and return to the display editor.

The main control menu selections function as follows:

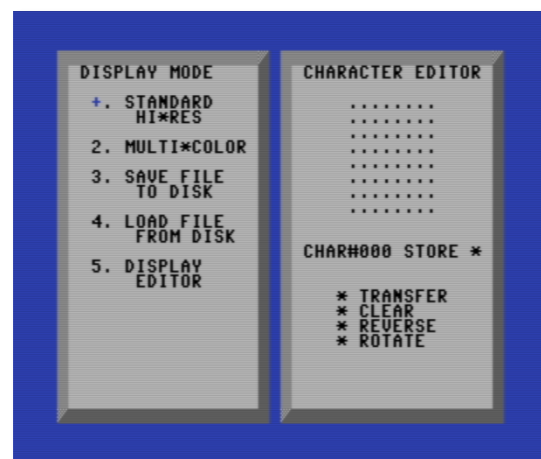
1. STANDARD HI-RES MODE...puts the display editor in standard display mode. Characters are displayed in single color high resolution.

2. MULTI-COLOR MODE... puts the display editor in multi-color mode. Characters are displayed in multi-color mode and horizontal resolution is cut in half. (Refer to the Commodore reference manuals for more information on multi-color mode graphics.)

3. SAVE FILE TO DISK...saves all data in the display editor including the character set to disk. When prompted for a file name enter any alphabetic name up to 10 letters. Only A-Z will be accepted. Use the delete key to correct any typing errors. Press Return when correct. Each screen area (see display editor section below) will be saved as a separate sequential file as will the color data for that area. Then the character set will be saved to disk.

Although this is a time consuming process it is necessary to allow transferability of data to user programs. Filenames for the display data are created as follows:

FILENAME+"CHC"+screen area number. The filename is the name you typed in and the CHC stands for character codes. The screen area number is from 1-16 since the program allows up to 16 screens to be created at a time. The color codes for each screen are saved in exactly the same manner except "COC" is appended to the filename instead of "CHC." The character set is saved as a PRG file. It can be identified by "CHARS" appended to the end of your filename. Refer to the section on transferring display data to user programs for more information.



4. **LOAD FILE FROM DISK...** loads a previously saved file from disk into the display editor. Enter the filename in the same manner as with the **SAVE** option when prompted. Enter only the original filename used with the **SAVE** option without appended characters.

NOTE: Filenames for the save option must be unique. No other files with that name can exist on your workdisk. Although the file will seem to save correctly the drive light will flash and no data will be saved. An error message may be displayed if there is a serious problem. These are as follows:

#4 file not found
#5 device not present
#8 file name missing
#9 illegal device number

5. **DISPLAY EDITOR...** brings up the display editor. (see section on display editor)

The Character Editor

The character editor is on the right side of the main control menu. To use it move the cursor over the 8x8 dot grid at the top and press the joystick button. The grid represents an enlarged character. Near the bottom right corner of the grid is an actual size representation of the character. Pressing the button will toggle that pixel. If it is on it will turn off and vice versa. This eliminates the need to select "draw" and "erase" modes as with other editors and speeds up the design process. With a little practice you should find this a very quick way to create the desired pattern.

When you achieve the desired pattern place the cursor over the "#" in the "CHAR#000" and press the fire button. Then move the joystick up or down to raise or lower the number. When you get the number you want for your character, press the button again and your cursor will return. Then place it over the "*" after "STORE" and press the button to store that character in your character set. It's that easy

Below the editor grid are four tools that will help make your job easier. These are:

TRANSFER... transfers a copy of the character whose number is selected into the editing grid for modifications or duplication.

CLEAR... clears the grid.

REVERSE... inverts the grid.

ROTATE... rotates the character currently in the grid 90 degrees clockwise.

NOTE: Modifications do not become final until stored.

The Display Editor

The display editor is where you put together your characters to create graphics displays. You can select either Standard or Multi-color mode from the main control menu at any time. Keep in mind your graphics will change depending on the mode you're in.

The display editor screen is divided exactly in half. The left side is the construction window and the right side is the control area. The control area consists of a blue border with an empty rectangle in the middle. This empty space is where your character set will be located. As you create characters in the character editor they will be stored here for use in the display editor. To return to the main control menu or character editor simply move the cursor to the border directly below the character area and press the button. This is a much faster way to move back and forth instead of having to fumble around for a small icon to make the switch. Anywhere under the character area will do.

The left side of the screen is the construction window. It is 20 columns by 25 rows in size, exactly half a display screen. To construct a display simply move the cursor over the character (on the right side of the screen) you want and press the button. A copy of that character will attach to your cursor.

Now move it into the construction window and press the button to place it in your display. Holding the button down while moving the cursor will repeat the character. Colors for both display modes can be changed by using the function keys as follows:

- f1... increment character color
- f2... character color
- f3... construction window
- f5... increment background #1 color
- f6... decrement background #1 color
- f7... increment background #2 color
- f8... decrement background #2 color

NOTE: It is recommended that the function keys only be used while the cursor is in the construction window. If they are used while the cursor is outside the window, the keystrokes are left in the keyboard buffer and not executed until within the window. This could result in loss of data or unwanted color changes.

Up to 16 separate or continuous screens can be created at one time. Since the construction window displays only half a screen at a time it can be scrolled to any section of the entire 16 screen construction area. This is ideal for creating huge scrollable backgrounds for games, maps, and so on. Since all 16 screens are always in memory and instantly available. The screens are arranged in the construction area as follows:

1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

The construction window is initially set on the left half of screen 1. Scrolling right 20 columns will place it on the right half of screen 1. Then scrolling down 25 rows will place it on the right half of screen 5 and so on.

To scroll the construction window move the cursor to the blue border anywhere above the character area on the right side of the screen and press the button. The cursor will disappear

and you can now scroll the window up, down, left, and right using the joystick. Pressing the button again will make the cursor reappear and you can proceed as normal.

Data Transfer Utility

As was mentioned earlier, the best feature of the *Displaymaker* utility is the ability to use the graphics you have created in your own programs. Now anyone can create eye-catching graphics for their own programs easily and quickly.

The only drawback to this approach is that everybody's needs are different. Since it's difficult to anticipate them, I have taken a generic approach to the problem.

I'm assuming that anyone who uses *Displaymaker* knows at least some BASIC programming if not machine language skills, so you should have no problem following the next few paragraphs.

When you save a *Displaymaker* file, the program actually saves 33 files to your work disk. This may seem like overkill but it is necessary to allow independent access to each screen of graphics.

The first 16 files are the character codes used to poke screen memory. Each file contains 1000 bytes arranged exactly as screen memory. The next 16 files are the color codes corresponding to the character codes arranged exactly as color memory. These are SEQ files. The final file saved is your character set used to create your screens. It is saved as a PRG file.

As I said I couldn't possibly anticipate everybody's needs, so much of the responsibility is on the user to make certain the data gets to the right place for use in programs. I have included a transfer utility to help speed the process and make it as painless as possible.

Probably the easiest way to use display graphics within a program is to have graphics data stored on disk and load the entire thing in with two LOAD commands; one for the character and color codes and one for the character set. Once in memory you can move it in and out of screen and color memory as you wish. This is the basic idea behind the *DataLoader* utility. It allows you to read in the sequential data files and configure them to suit your needs.

When you have it where you want it, you can save it as a program file without having to know anything about file programming. You can also move your character set to the most convenient location and resave it to disk so it will always load right into place. Here's how to do it.

Reset your computer and put in the *Displaymaker* disk. Type LOAD "DATALOADER",8 and press Return. When the program has loaded, RUN it and a title will appear along with a FILENAME prompt. Enter the filename you wish to load along with it's appended characters (e.g. "GRAPHICSHC01") and press Return. (Remember you must transfer each file separately. However many screens you create, you must transfer both character and color code files for each screen).

The next prompt is the LOADADDRESS. This is the address at which you want to begin storing data for that screen. The next two prompts are the ROW LENGTH and STEP SIZE. These are used to configure your data. Normally the ROW LENGTH will be 40 for a single full screen display but can be 1-255. The STEP SIZE is mostly used for large multiple screen

displays such as game backgrounds. A value of 40 for the STEP SIZE and ROW LENGTH will load 1000 consecutive memory locations arranged like screen memory.

The STEP SIZE value is added to the beginning address of each row to point to the start of the next row. It sounds more complicated than it is. I usually write down where everything should go before I start. Things go a lot smoother and faster this way.

When the transfer is complete, you will be prompted for the next filename. Keep repeating the process until all files are in the correct locations.

Now you are ready to save the whole thing as a single PRG file. At the FILENAME prompt enter "SAVE" and press Return. You should see the "SAVEASPRGFILE" title and a prompt for a filename. Enter the name you want.

Next enter the start address for the save, then the last address to save plus one. The program will save all memory from the start address up to but not including the +1 address. In other words to save up to 49152 you enter 49153. (the plus 1 is necessary because of the way the save routine works). Make certain you have the correct disk in the drive before you press Return.

The character set you created with *Displaymaker* automatically saves to your work disk as a PRG file. It's load address is always 8192 (\$2000). If this address is not convenient for you, it can be relocated. Simply enter "CHARS" at the main FILENAME prompt. Then enter the new address when prompted and a filename. (You must enter a filename even if it is the same as the old one. But the new file must be on a different disk if it is going to have the same name).

Once the character set has been loaded to the new location a PRESSANYKEYTOSAVE prompt will appear and the program will wait. Make sure you have the correct disk in the drive and press a key to save the file.

Now your ready to go! All your graphics data can be loaded back in with two simple load commands: LOAD "CHARSET",8,1 and LOAD "GRAPHICS",8,1. Be sure and include the ,1 extension to load to the correct address.

Very Important

In order to allow as much free memory for data as possible, the *DataLoader* utility has no error trapping built in. The user must be sure to type in only valid decimal memory addresses and unique filenames. Errors are normally not fatal, but could inadvertently destroy important data. Just be sure to go back and redo anything that could have been corrupted.

The available workspace for the transfers is from 4096(\$1000) to 53247(\$CFFF). However, BASIC programmers can only use up to 40959(\$9FFF) because the BASIC interpreter is needed to run your programs. The memory is useable in the *DataLoader* utility but BASIC programmers have no means to access the extra memory and still keep BASIC active.

If an attempt to transfer data to an address below 4096(\$1000) is made the *DataLoader* program will crash. This is where the basic and machine language routines are stored. If it is desirable to locate data below this point a machine language monitor can be used to transfer the data to the new location and save it.

Machine language programmers can fully utilize all memory by switching out the BASIC interpreter for 8K RAM. The SAVE routine in the transfer utility temporarily switches out BASIC while saving. In this way the entire work area is available for storage.

Finally, it is up to both the BASIC and machine language programmers to set the screen, border, and background registers 1 and 2 from within their programs when displaying graphics created using *Displaymaker*. Just write down your color combinations while using *Displaymaker*, then look up the color codes for poking screen, color, border, and background registers 1 and 2.

That's about all there is to it. The best way to learn is just dive in and have fun. You'll be amazed what you can do so easily!

Mapmaker

David Pankhurst

Use this utility to create and manipulate line drawing maps of the world. You can also use it to track hurricanes.

Mapmaker lets you create and manipulate line drawing maps of the world. You do this by entering points; a series of points form a line, and a number of lines can form the outline of a state, a travel route, a storm's path, or whatever.

The maps are easily editable, and maps can be viewed in any of the 64's 16 colors. You can view several maps simultaneously and see them from any view area and scale.

To start, load and run *Mapmaker*. Upon starting, it checks for any maps already in memory. This is handy if you quit the program for any reason, and then rerun it. Eventually a menu of options is displayed. Some are obvious; QUIT exits the program, CREATE makes a new blank file, and ERASE deletes a file from memory (the maximum you can have in memory at any time is 21 files of 227 records each). The LOAD option lets you work on a previously saved file from disk. SAVE places the resulting file on disk. Note that *Mapmaker* files are specially encoded, so other files cannot be used.

The EDIT option is where most of the work is done. To work on a file, select EDIT, and choose from the listing of files in memory (If you want to work on a new file, first use the CREATE option and then EDIT.)

After you select a file, it is listed to the edit screen. The up/down cursor keys allow you to scroll through the list. All operations affect the current entry, which is the one indicated at the top of the screen. You can delete it with the Delete key, or insert a new record at this point with the Insert key. And pressing the Return key allows you to edit that entry.

Map coordinates are entered by the latitude and then longitude of the location. Latitude ranges from 0-89 degrees North or South of the equator; longitude is 0-179 degrees East or West of the Prime Meridian in Greenwich, England.

Entries are made first for the degrees latitude, then the minutes (60 minutes equal 1 degree), and the direction north or south of the equator. Following is longitude, with degrees, minutes,

and east or west. If an entry is too large (for example, 95 degrees of latitude, or 67 minutes) you must enter again. When all of the line is correctly entered, the line is reprinted, and the other edit keys are again available for use.

Adding entries can be done by using the Insert key to add a record, then Return to edit it. However, an easier way to append records to the end of a file is by cursoring to the end of the file (the record marked -----eof-----) and pressing Return. A new record is displayed, and you can immediately edit it.

When editing of the file is finished, use Shift/Return to save the file, and return to the main menu.

Normally, all entries in the edit screen are map coordinates. However, several commands are available for adjusting the map display. These are entered in place of the first entry when editing a line (the degrees latitude), and are the C, D, PU, and PD commands. The D command switches drawing pens, or color, for drawing. The map display has three color selections, or pens, available on screen at one time, called pen 1 through pen 3.

The four possible commands are: D0, D1, D2, and D3. All subsequent drawing takes place using the selected pen. D0 is only used to set the background color. If you set it, and then try to draw, the lines will be the same color as the background, and hence invisible.

The current pen is assigned a color with the C command. It follows the format of C<number>, where the number ranges from 0 to 15, representing a Commodore color. The following table shows the choices:

- 0-Black
- 1-White
- 2-Red
- 3-Cyan
- 4-Purple
- 5-Green
- 6-Blue
- 7-Yellow
- 8-Orange
- 9-Brown
- 10-Light Red
- 11-Dark Grey
- 12-Medium Grey
- 13-Light Green
- 14-Light Blue
- 15-Light Gray

When the C command is executed, the color of the current pen is changed to this color code. This affects all lines drawn with that pen number, both already on screen, and in the future.

Using the C and D commands together allow you to set all the available map colors. As an example, look at the following entries from a map file:

D0
C1
D1
C5
D2
C7
D3
C0

Here, the background color is selected, and then set to white (1). Immediately, the background color is changed. Then pen 1 is selected, and colored green (5). Following that, pens 2 and 3 are each in turn selected and set to yellow (7) and black (0), respectively. Since pen 3 was the last selected, all subsequent lines will be drawn in pen 3's color, black.

Two final commands that can be entered allow for drawing more than one long connected line. PU and PD are shorthand for Pen Up and Pen Down. When the pen is down, it is drawing with the current pen and color. When the pen is up, no drawing occurs until you execute a PD command. These commands allow you to draw complex shapes that are not made up of a single unbroken line, such as text, or marker symbols.

These command codes aren't necessary in files. If left out, the current color and pen is used. If a file is displayed and changes the colors or current pen, these changes become the defaults. For this reason, it may be advisable to select your colors or pens at the start of each file. Alternately, you could create a small file to set the colors the way you want at the start of each viewing session.

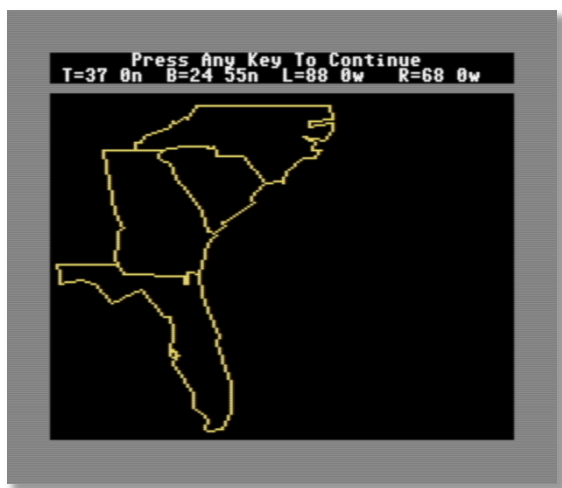
The final menu option to consider, VIEW, is where all your hard work becomes apparent. You are offered a list of all the maps in memory. Selecting the file by pressing its letter places a checkmark in front of it. To deselect it, just choose it again, and the check mark disappears. Pressing Return causes all checked files to be set up for display.

The display of the selected maps is on an adjustable viewscreen. The coordinates for each of the windows' four sides describe the viewing area. Defaults are provided, or you can enter your own choices.

The bottom coordinate displayed is a special value calculated to make the view proportionally correct. This is the computer's best fit as to how the world really looks at that view area. Accept it by pressing Return. However, you can change it, for instance to fit more onto the display, or to stretch or shrink a view.

Scaling also affects the map in other ways. The map display is rectangular, meaning that large scale displays will be distorted. A large map of the world with Antarctica drawn in, for instance, would extend across the whole bottom of the screen. To keep distortion to a minimum, view smaller areas, and use the computer's suggestion for the bottom viewscreen coordinate.

To abort drawing, and return to the main menu, press any key. Maps are drawn in the order they are listed in the menu, with the default colors, or the colors that were in effect from a previous map drawing.



To start you on your way, several map files are included: FLORIDA, NORTH CAROLINA, SOUTH CAROLINA, and GEORGIA. Each draws an outline of a state. Also included is BOX, which allows you to check the aspect ratio of your screen.

Some notes on customizing the program; Line 9100 has the colors for text: c(0) is the background, c(1) is pen 1, c(2) is pen 2, c(3) is pen 3, c(4) is the border color, and c(5) is the regular text color. Line 9090 has the aspect ratio AS. This can be changed if scaling isn't correct on your screen.

To check, load and select VIEW for file BOX. When entering the viewscreen side coordinates, accept all defaults, including the one for the bottom edge. The box displayed should be perfectly square. If not, adjust the variable AS. Increasing it will shorten the box height, and decreasing it will increase the height.

Mailing List

B. A. Carwithen

Keep track of names and addresses with this short mail utility.

Mailing List is a simple little program that I wrote in order to keep track of family names and addresses for Christmas cards. While it may be a little past that time of year now, this program can be useful for any other list of names and addresses that you may have.

It is written entirely in BASIC and should be easy enough to modify by the average programmer.

When you run the program, you'll see a menu with five options: List all last names, Search for a name, Add names, List all names and addresses, and Quit. Press the function key next to the option to make your selection.

Search For a Name

The idea behind the program is that you enter a person's last name and that person's name, address, and phone number appear onscreen. Press f3 to search for the name of your choice. At the prompt, enter the last name. (Wildcards are not accepted.)

If the name is located, it prints to screen. If two or more names in your database are the same, they will all appear onscreen. To print out the name and address, press the 0 key at the prompt. If the name isn't found, the program will inform you and return to the main menu.

List All Names

But what if you can't remember how cousin Barney spells his last name? In that case, press f1 to list all of the names in your database for the correct spelling. A sorting routine puts them in

alphabetical order. If you have more than 20 names in your database, they will appear one screen at a time. Simply press the space bar to continue to the next page. After the final page, you will return to the main menu.

List All Data

There are times when you might simply like to scroll through all the names in your database. In my case, I knew everyone in mine should get a Christmas card, so I simply had all of them to list to the screen, four at a time. Press f5 to list all the names, address and other data.

Data Statements

The names on the list are stored as DATA statements and are read with a READ command. For the program to operate properly, you need to enter eight data statements for each entry. The program reads data for last name, first name, address, city, state, zip or postal code, and telephone number. You must have eight items for each name when the program runs or you will get an OUT OF DATA ERROR in line 1140. As with all data statements, the items must be separated by a comma. If you are not sure of a zip code, telephone number, or other piece of data, simply enter a question mark, or some other blank data. Even a comma will suffice.

Enter Data

Enter your own names and addresses by pressing f4. This lists the data statements starting at line 500. Several sample names and addresses have already been entered as a convenience. Use lines 500 - 699 for your data. The program has been dimensioned in line 1110 to accept 200 names. Feel free to change this if necessary.

After you enter your own data, run the program again and test it. If you get an OUT OF DATA ERROR, list the lines from 500-700 and examine them for mistakes.

If you have long names and addresses and cannot get all eight items to fit on one data line, that is not a problem. You may use as many data lines as you need. Just be sure that you have eight items for each person.

Save It

When everything is running as it should and you are ready to quit, press f7 at the main menu. You will be prompted with a reminder to save your program. If you have added or deleted data, press Y for Yes at the Save prompt. The program will then scratch the old version off your disk and save the new one.

Caution: Do not try to save your own data on the *Gazette Disk*. It is write protected and your data will not save. Load the program and then save it to a formatted work disk.

If you find that you've made an error by trying to save the program to the wrong disk, simply type RUN again. The program should still be in memory. Insert the correct disk and press f7 to save the program again.

Hurricane

David Garner

Enjoy this short colorful graphic adventure game.

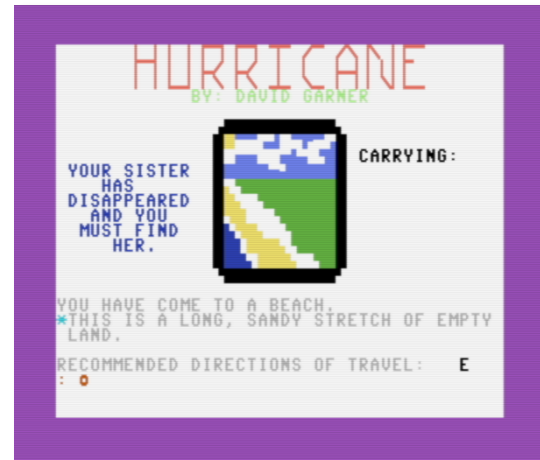
Your sister is missing and you have to find her. That's about all I am going to tell you about this colorful little graphic adventure game, and the fact that it varies with each play.

After you've played it a few times, you can sit back, relax, and roam around the different screens.

When you run *Hurricane*, it will load a file called GRAPHICS and the screen will change colors. You will find yourself on a beach. The only command the game will understand on this screen is SEARCH. Type it and press Return.

Each screen has its own command word. Other words include LOOK and THROW. You can also use the letters E, W, N, and S to indicate direction of travel. If you use a word with which the computer is not familiar, it will ask "What?"

Note that in one mode in which the game sets up, play is especially short.



PD Picks on Disk

Ecstatic Code

Cosmos Designs

This program is a demonstration of what the 64 can be made to do in the hands of a talented programmer. Simply run this program and check out its special effects.

(According to the C-64 Scene Database, currently hosted at csdb.dk, Cosmos Designs was an Austrian demo and game dev group. This particular demo is called "Ecstatic Code" and was produced for the TAT & Magnetix Copyparty held in 1989. The demo, one of three submitted by the group for this event placed second among four entrants. -MS)

Warriors

Paul Spomer, Marc Turner, and David Turner

This graphic adventure has its own instruction program on the flip side of this disk. When you become familiar with the game, you can simply load and run WARRIORS to save time.

(The full name of this game is Warriors of the Crown. According to Gamebase 64, this game was created by Paul Spomer, Marc Turner, and David Turner using SEUCK—the Shoot 'Em Up Construction Kit—from Sensible Software.)

For more information about these programs, see the "PD Picks" column in this issue.

If you have missed any issues of Gazette Disk for 1994, they are available for \$11.95 each. This price includes shipping & handling to the U.S and Canada.

LAST CHANCE!!

The following offer expires January 31, 1995!

FREE DISKS!

For Gazette Disks prior to 1994, we have a great bargain for you! You can have any issue of Gazette disk for free — you just pay a shipping and handling fee of \$3.50 per disk!

We have most disks in stock from June 1986. Just remember, these disks do not have instructions. You will need the magazine in order to utilize the programs to their fullest.

FREE MAGAZINES, TOO!

You may have back issues of COMPUTE or Gazette magazine for a shipping & handling fee as follows:

1 magazine	\$3.00
2 magazines.....	\$3.75
3 magazines.....	\$4.50
4 magazines.....	\$5.25
5 magazines.....	\$6.00

For 6 or more, add \$.75 for each additional issue. No express orders, please. All checks or money orders must be in U.S. funds. Canadian and all other foreign orders, please add an additional \$5.00 for postage. Not all issues of these magazines are available.

Send orders to

COMPUTE Publications
324 W. Wendover Ave. Ste. 200
Greensboro, NC 27408

Be sure to specify the month and year of the issues that you want. For more information, call (910) 275-9809 and ask for Customer Service from 9-4 ET.

NOTE: All orders must be received by January 31, 1995.

Not all issues are available, so you may want to contact our office by telephone. Supplies are limited. Please allow four to six weeks for delivery.

Gazette Special Disk Library

Here are some of the special 64 and 128 collections that are available from COMPUTE Publications.

SPEEDSCRIPT

\$13.95

COMPUTE Publications' most popular program ever. Powerful word processing package includes SpeedScript for the 64, SpeedScript 128, spelling checkers for both 64 and 128 versions, plus an additional dozen support programs, including mail-merge and word-count utilities.

GAZETTE'S POWER TOOLS

\$9.95

Fourteen of the most important utilities for the 64 ever published in Gazette. For serious users. Titles: MetaBASIC, disk Rapid Transit, Mob Maker, Ultrafont+. Quick!, Disk Editor, Basically Music, PrintScreen, 1526 PrintScreen, Fast Assembler, Smart Disassembler, Comparator, Sprint II, and Turbo Format.

BEST GAZETTE GAMES

\$9.95

Best Dozen arcade and strategy games ever published in Gazette all on one disk. All games for Commodore 64. Titles: Crossroads II: Pandemonium, Basketball Sam & Ed, Delta War, Heat Seeker, Omicron, Powerball, Q-Bird, Trap, Arcade Volleyball, Mosaic, Power Poker, and Scorpion II.

GAZETTE INDEX

\$9.95

Every article and department from Gazette--July 1983 through December 1991 issues--is indexed: features, games, reviews, programming, BugSwatter, Feedback, and the other columns. Disk features pull-down menus, help screens, superfast searching/ sorting capabilities, and much more.

THE GEOS COLLECTION

\$11.95

Gazette's best 13 programs for GEOS and GEOS 128 users. Selection includes utilities, applications, and games. Titles: Super Printer Driver, Skeet, File Saver, Help Pad, Word Count, Directory Printer, Quick Clock, SlideShow, File Retriever, Screen Dumper, Font Grabber, GeoPuzzle, and GeoConverter.

128 CLASSICS

\$11.95

Thirteen of Gazette's best 128 programs, including utilities, games, and applications. Titles: MetaBASIC 128, RAMDisk 128, 80-Column Disk Sector Editor, MultiSort, Block Out, Miami Ice, The Animals' Show, Cribbage, XPressCard, Sound Designer, Video Slide Show, Math Graphics, and 3-D BarGrapher.

Order all six disks for \$49.95. Prices include shipping and handling. Send check or money order to

Gazette Disks
COMPUTE Publications
324 W. Wendover Ave., Ste. 200
Greeneboro, NC 27408

You may pay by credit card for orders over \$20. Be sure to include the card type, number, expiration date, your signature, and a daytime telephone number.

Residents of NY, NC, PA add appropriate sales tax. For deliveries outside the US or Canada add \$1 for surface mail, \$3 for airmail.

It's still available!

MAPPING THE 64 & 64C
by Sheldon Leeman

This classic 64 reference book is a comprehensive memory guide for beginning and advanced programmers.

To order your copy, send \$18.95 plus \$2.50 for shipping and handling to

COMPUTE Books
324 W. Wendover Ave., Ste. 200
Greensboro, NC 27408

(In Canada, send \$4 shipping and handling, \$6 in other countries. All orders must be paid for in U.S. funds.)

—— MODEM SALE ——

COMPUTE magazine has a limited number of 9600 bps modems for sale at a great price. If you have an IBM-type PC, you can get one of these U.S. Robotics internal modems for only \$49.95, plus \$5.00 shipping and handling.

—— \$49.95 + \$5.00 ——

Send check or money order for \$54.95 to

COMPUTE Modem Sale
324 W. Wendover Ave. Ste 200
Greensboro, NC 27408

Order by credit card by calling
Customer Service at (910) 275-9809.

Risers And Sliders Arcade Action game
50 levels of play.

Now only \$6.00 Plus Free S&H in USA

Attention: C64/128 User Groups:

Ask about our New Rates.

New P.D. and Shareware Disk. Packed!

Games *Utilities* *Data Files* More!

Send \$3.00 for Disk and S&H costs.

MicroStorm Software (R)

Box 1086 Sidney Mt. 59270